Improving performance of a GSM-Based Speech Recognizer

Samson Lupembe

Department of Electrical Engineering
University of Cape Town
Cape Town
South Africa

Submitted in fulfillment of the requirements for the Master of Science degree in Electrical Engineering

August 2004

Declaration

I declare that this dissertation is work undertaken by myself. I am submitting it for consideration to be admitted to a Master of Science degree in Electrical Engineering at the University of Cape Town. To my knowledge, it has not been submitted before for any degree or examination at this or any other University.

Signature: _____
Samson Lupembe

Acknowledgments

First I would like to thank god for the wisdom and strength to complete this work. Thank you Dr. DJ Mashao (Supervisor) for your willingness to help whenever I needed help. Most important my love Teresa Mausse (Tete), thank you for being such an inspiration to me. Thanks to my fellow students Nicholas (Pj), Leno, Ofentse, Lerato, Limpho, Dirshan and Francois for the support. Last but not least my family for your added support.

Abstract

The accuracy of speech recognition systems degrades when operated in Global System for Mobile Communication (GSM) channels. The degradation is mainly due to channel effects which put limitations on the use of speech recognition applications over GSM networks. Hence, if speech recognition technology is to be used over GSM networks, recognition accuracy of GSM channel speech has to be improved.

In recent years, many channel compensation techniques have been developed in an attempt to improve recognition accuracy for GSM speech. There are generally three classes of channel compensation techniques; filter based (pre-processors), feature modification and model modification techniques. Filter based and feature modification attempts to reduce effects of distortions prior to the recognition process while model modification attempt to reduce effects of distortions arising from the channel during the recognition process. In most cases these techniques are used together in a recognition system.

This study begins with an investigation of two channel compensation techniques. These are the Cepstral Mean Normalization (CMN) and Vocal Tract Length Normalization (VTLN). The study applied both the CMN and VTLN in an attempt to improve the performance of speech recognition on GSM speech. The results show that the channel compensation technique yield improvement of about 35.9% drop in word error rate (WER) compared to the case where no channel compensation is used. This system is used as the baseline system in this study.

A further contribution toward improving the robustness of speech recognition systems to GSM speech, investigates the use of the speech enhancement noise suppression filter (as a speech recognition system pre-filter). This filter is addition to the channel- and speaker normalization techniques mentioned above. It was found that the filter enhances speech recognition robustness to GSM speech by reducing the channel effects on GSM speech signal. Experiments conducted proved that the use of the proposed pre-filter does improve recognition performance by 40.2% drop in word error rate (WER) compared to the case where no channel compensation is used at 90% confidence level.

Contents

List of Figures

List of Table

Part 1

Background

Chapter 1

Introduction

Communication between human beings is important and the most effective way of passing information. Humans are also able to communicate with machines for instance, computers, where keyboards and typing are the means of communication. Most people can speak but not everyone can read or write. Therefore, if we could get the machines to understand human speech, we could be able to communicate with people (and even make communicating with computers open to many people). This is the main motivation behind Automatic Speech Recognition (ASR) as a field of research, to enable machines to recognize human speech. Automatic Speech Recognition may then be defined as a process carried out by a machine to extract information contained in a captured acoustic speech signal and converting it into words.

Research in the field of speech recognition has been done for almost four decades [87]. However significant progress and notable breakthroughs in the field were only made in the last two decades. Today we have numerous commercial speech recognition products such as Dragon Naturally Speaking offered by Scansoft and IBM's Via Voice [113]. Even with the widespread commercialization of speech recognition products the technology still has a number of shortcomings. One of the main challenges is to overcome varying performance of speech recognition systems with acoustic environments. Environment in this case refers to any medium, characterized by its acoustic properties, through which a speech signal travels to get to speech recognition

engine. In general, performance of speech recognition systems degrades when they are used in different environments to those in which they were trained, this is also known as a mismatch in training and testing conditions. Despite the fact that these shortcomings have not yet been fully overcome, speech recognition has advanced enough to be applied in practical applications, for instance desktop dictation software. Other applications would be accessing remote recognition engines over certain communication networks like GSM (Global System for Mobile Communication).

Currently, GSM networks present an acoustic environment (GSM voice communication channel) that produces poor speech recognition. As mentioned above, one of the main challenges in speech recognition is to attain recognition performance that is unaffected by acoustic conditions of use. Hence in this study, we focus on reducing the effects of GSM channel on the quality of speech signals to improve recognition rate of GSM-Based speech recognition systems. There are three possible modes of operation for using speech recognition over GSM networks:

- **Terminal Speech Recognition.** In this modality, recognition takes place on the mobile device itself e.g. a cell-phone. Because of high computation power and memory requirements of speech recognition this approach is only limited to less sophisticated tasks such as voice dialing.

- **Distributed Speech Recognition.** In this modality, the speech recognition process is distributed across different points on the network. One such approach is having a terminal/mobile device carrying out feature extraction process on speech signal and transmits extracted features over the network to a remote recognition engine. The main disadvantage of this modality is that it is costly.

- **Network Speech Recognition.** In this modality, a speech signal propagates through a mobile network to a remote recognition engine. The main advantage of this modality is that it allows for a powerful computer in remote server to perform sophisticated recognition operations that cannot be performed on the mobile

device like a cell-phone. However, the main disadvantage is that it is exposed to channel imperfections such as transmission noise and data loss.

In this study, a Network Speech Recognition modality is adopted with a view to correct its disadvantage. Hence the study attempts to reduce these negative effects with the aim of improving robustness of recognition systems to GSM speech. The issues causing degradation of GSM speech that lead to poor performance are discussed in the following section.

1.1 Causes of Degradation of GSM Speech

The poor performance of speech recognition systems on GSM speech is mainly caused by the speech-coding scheme employed in the GSM standard (The Regular Pulse Excited with Long Term Prediction (RPE-LTP)). The objective in speech coding is to represent speech with a minimum number of bits while maintaining its perceptual quality. Speech coding allows more efficient use of the available bandwidth for the purpose of efficient transmission or storage [17].

In general, speech coding degrades the quality of a speech signal. The reduction in bit rate itself degrades performance of Automatic Speech Recognition (ASR) systems, in addition certain coding schemes (especially low-bit speech coding schemes) introduce distortions to speech signals thereby lowering the signal quality even further which is the case with the speech coding scheme used in GSM. Previous research has shown that recognition performance of ASR systems drops with dropping bit rates [17]. One of the conclusions drawn by Lilly and Paliwal [17] is that bit rates above 16kbps display good recognition performance which means Public Switched Telephone Networks (PSTN) do not affect ASR performance much in terms of speech coding since their bit rate are above 16kbps. Examining the GSM full-rate speech codec, which operates at 13kbps, and the half-rate codec at 5.6kbps, shows that the lower bit rate is one of the major reasons why ASR over GSM mobile network is poorer than that over PSTN network.

In addition to the effects due to coding, the GSM channel noise contributes to the degradation of speech signals over the GSM network (termed GSM speech). Usually, speech transmitted over a communication channel is often expressed as,

$$y(t) = s(t) * h(t) + n(t) \tag{1.1}$$

Where * is convolution, $h(t)$ is the channel transfer function, $s(t)$ is the transmitted speech signals and $n(t)$ is the channel additive noise. Equation 1.1 can be used to express a GSM channel speech. If $h(t)$ and $n(t)$ were definitely known they could be compensated for and speech recognition performance with such speech would be kept high enough but the problem is that these two are often not known and vary with time. A number of channel compensation techniques such as Cepstral Mean Normalization and Vocal tract length normalization, which attempt to alleviate the effects of channel noise on speech signal, have been developed and are mentioned and discussed later.

1.2 Previous Related Work in GSM Speech Recognition

Literature reviews considerable research work in speech recognition in GSM environments. AURORA, which is European Telecommunication Standard Institute (ETSI) Working Group, is the leading research group in this field, with a focus on the Distributed Speech Recognition technology. Apart from the work done by AURORA, some journals and postgraduate research relating to resolving these issues have been undertaken [17, 103, 104, 105, 106, 107]. Lilly and Paliwal [17] studied the effects of speech coding on ASR performance. Euler and Zinke [103] studied the effects of speech coding on recognition performance and went on to analyze the performance of different cepstral features recognition. Other research work efforts included an attempt to perform recognition from GSM codec parameters assuming availability of such parameters during recognition [104].

Huerta and Stern [105] proposed a classification method aimed at reducing the degradation in recognition accuracy by full-rate GSM codec. Gupta in [106] used a two-level cepstral mean subtraction robustness approach to improve robustness of ASR systems to GSM environmental noises. Soulas in [107] approached the problem by adapting PSTN models to the GSM environment using spectral transformation. Karray [108] proposed robustness HMM (Hidden Markov Models) architecture for impairments, which are encountered in GSM cellular network. Chang [109] tested the readiness of the current commercial ASR systems for development over mobile networks such as GSM network and concluded that, simple small vocabulary task specific systems are robust enough to mobile communications network effects. However, more complex systems with large vocabulary and high perplexities showed very low levels of robustness to mobile communication environments.

Research was carried out in [110] with a focus on preprocessing and HMM parameter adaptation techniques to improve ASR robustness for PSN (Packet Switched Network) and GSM speech. The preprocessing methods investigated were cepstral normalization, high-pass IIR filtering of cepstral trajectories and blind equalization based on adaptive filtering which was done with real GSM and PSN data with ASR systems with 50 words in their vocabularies. In this study, a speech enhancement noise suppression filter is investigated as a preprocessing filter for robust GSM speech recognition. Real GSM data is used in this study for testing purposes with the ASR systems of about 6,000 words in their vocabularies as will be seen in Chapter 5.

1.3 Thesis Objectives

This study begins with an investigation of two channel compensation techniques, the Cepstral Mean Normalization (CMN) and the Vocal Tract Length Normalization (VTLN). The CMN and VTLN are employed in an attempt to improve the performance of speech recognition on GSM speech.

As a further contribution toward improving the robustness of speech recognition systems to GSM speech, the use of speech enhancement noise suppression filter (as a speech recognition system pre-filter) is investigated. This filter has been used in combination with the channel- and speaker normalization techniques mentioned previously. The filter enhances speech recognition robustness to GSM speech by reducing the channel effects on GSM speech signal.

The main aim of this study is to address the questions:

1. Does the use of noise suppression filter as pre-filter improve robustness of recognition systems to GSM speech?

2. Does improving perceptual quality of GSM speech necessarily yields improvements in recognition of such speech?

The approach taken in this study was to test recognition systems, which don't employ any normalization technique with GSM speech to determine their performance on such speech. The well-known channel normalization technique, Cepstral Mean Normalization (CMN), was added to these recognition systems to determine improvements it would bring to performance of these systems. Then speaker normalization technique, Vocal tract length normalization (VTLN), was added to these recognition systems as well to determine improvements it would bring to performance of these systems. Having determined the improvement brought by the combination of channel- and speaker normalization, the noise suppression filter is implemented to see if it brought about any further improvements in performance to this combination. Since the filter under investigation is widely used in enhancing noisy speech and useful in the area of low-bit rate digital communication, with the motive of improving perceptual quality of the speech, this study seeks to find out whether improvement in perceptual quality of speech necessarily brings improvements in recognition of such speech in general.

In particular, a front-end used by the CU-HTK (Cambridge University Hidden Markov Toolkit) group [96, 97] including CMN and VTLN was used for base-line results and evaluation of the effectiveness of the noise suppression filter under investigation in this study. All systems used were built using Hidden Markov Model Toolkit version 3.2.1 (HTK v 3.2.1) following a procedure outlined in Chapter 4.

1.4 Thesis Outline

The thesis is outlined as follows. In Chapter 2 we describe the general speech recognition concepts covering both the front-end, which perform the necessary signal processing on the speech and the back-end, which involves pattern classification.

Chapter 3 covers the concept of Speech coding schemes including its application on GSM since Speech coding and compression is an important element in efficient transmission of speech over digital wireless systems.

Chapter 4 describes the experimental setup and procedure followed. This chapter describes how recognition systems tested were built including the base-line results against which the effectiveness of the noise suppression filter under investigation will be evaluated.

Chapter 5 provide the theory of the two tested Normalization techniques, Cepstral Mean Normalization and Vocal Tract Length Normalization, together with the results of experiments conducted with them in order to determine their performance on GSM speech and hence determining the baseline.

Chapter 6 discusses the theory of the noise suppression filter proposed in this study as well as the results obtained from experiments conducted with it.

Chapter 7 summarizes the study and outlines the conclusions drawn from the results obtained from experiments. On the basis of the summary and conclusion further research work is suggested.

Chapter 2

Speech Recognition Concepts

This chapter describes the general speech recognition concepts. A general model for a basic speech recognition system, shown in Figure 2.1, is made-up of two major sections known as the front-end and back-end. The front-end performs all necessary signal processing on speech signal whilst the back-end involves some method of pattern classification (during training) and acoustic pattern recognition (during testing). In the following sections, some of the basic speech recognition concepts are discussed.

2.1 Front-Ends: Speech Signal Processing

The ultimate goal of signal processing front-end is to extract features that are consistently constant in a speech signal and use them in recognition systems. Speech recognition is not performed directly on digital speech signal fed into a speech recognition system. This is done because there are many source of variability associated with speech signals. Some of the well-known sources of these variations are due to the individual speaker (e.g. different lengths of speaker's vocal tracts), environments of use (e.g. background noise), microphones used for speech capture and pitch of voiced segments of spoken words. Other names for speech signal processing include front-end processing, feature extraction and signal modeling.

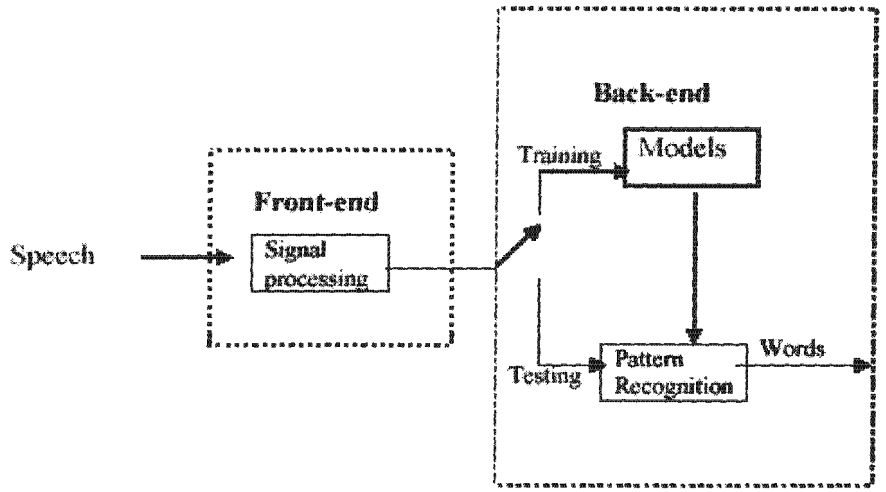


Figure 2.1: Basic Speech Recognition System block diagram

The process of speech processing involves converting the recorded analogue speech signal into feature vectors that can be applied in any speech processing application of the user's choice. This generally involves the following three steps, namely, analogue to digital conversion, preprocessing and feature generating.

2.1.1 Analogue to Digital Conversion (ADC)

Analogue to digital conversion (ADC) is important because it converts analogue signal to digital. Speech when first captured is an analogue signal and yet a computer which processes the speech is a digital device. Therefore the analogue speech signal must first be converted to a digital signal that can be discretely manipulated for signal processing. This is done by an analogue to digital converter (ADC), which is usually a specific component inside either a digital signal processing (DSP) card or a sound card in a computer. ADC can be subdivided into three steps, namely, sampling, quantization and coding.

Of the theorems in communication theory, the most applied currently is the time-domain sampling theorem. The reason for the theorem's importance is that the digital computer is a powerful and commonly used tool for signal processing, and since we live in a world that is basically analog, some guideline are necessary to allow analog signals to be digitized without loss of information. These guidelines are specified by the time-domain sampling theorem.

Quantization is the process of converting a sampled continuous analogue speech signal into a digital signal by expressing each sample as a finite number of discrete amplitude. The coding process of assigning a unique binary number to each quantization level follows this. Examples of such coding techniques are the uniform quantizer and pulse code modulation.

2.1.2 Preprocessing

Preprocessing constitutes the speech framing, windowing of the framed speech signal and the pre-emphasis. Speech goes through a preprocessing stage after the analogue to digital conversion in order to improve the performance of the recognition system and prepare the speech for the feature generating.

Preprocessing is performed on short windowed frames of speech, often 20 to 25ms at some frame rate, often 10ms. The most popular window function in speech recognition is the Hamming window. Window function is defined as,

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right), & 0 \le n \le N \\ 0, & otherwise \end{cases} \qquad (2.1)$$

where N is the length of the window in samples.

2.1.3 Pre-emphasis

Pre-emphasis filter is frequently applied to emphasize high frequencies in a speech signal prior to feature extraction. It is typically a simple first order high pass filter that increases the relative energy of the higher frequency spectrum. The transfer function of the pre-emphasis filter often used is expressed as,

$$H(z) = 1 + az^{-1} \tag{2.2}$$

where a is a constant normally in the range of (-1, -0.4) [90]. Voiced sections of speech have a spectral slope that is tilted to the right (or a negative slope) such that high frequencies are de-emphasized. By applying a pre-emphasis filter these frequencies are boosted. It is noted that it doesn't really matter if pre-emphasis is done before or after windowing [89].

2.1.4 Speech features

Once pre-emphasis and windowing is completed, from each frame the features that have been computed are put in feature vectors. With feature extraction performed on each windowed segment of speech, a complete signal will be parameterized into a sequence of feature vectors at the end of the feature extraction process.

Speech signal processing methods are usually classified into three broad categories, namely, Linear Predictive Coding (LPC) feature extraction, filterbank based feature extraction and auditory base methods feature extraction [87]. Each of these classes is discussed in the following sections.

2.1.5 Linear Prediction Coding (LPC) feature extraction

LPC modeling was first introduced in the early 1970s [92]. Since then it has been widely used in speech processing fields including speech coding, speech synthesis, speaker recognition and verification, speech recognition and many other speech related technologies.

The LPC coefficients are derived from the speaker's vocal tract model [89]. They therefore capture physiological information from the voice of the speaker. The concept behind linear prediction coding is to model the pre-processed sample in a speech signal as a linear combination of its previous samples values on the signal. This is expressed as follows,

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{2.3}$$

where p, known as the LPC model order, denotes the number of previous samples used to model the current or the nth sample, G is a gain scaling factor, $u(n)$ is the present input and a_k are coefficients of the linear combination known as linear prediction coefficients. In speech applications, the input $u(n)$ is usually ignored.

The difference or error between a signal and its LPC model is expressed in an equation form as,

$$e(n) = s(n) - \hat{s}(n) \tag{2.4}$$

The z-transformation of Equation 2.4 yields,

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)} \tag{2.5}$$

where A(z) is known as the *p*th-order inverse filter. Optimization of LPC analysis is achieved by minimization of the mean square prediction error (MSE) [93], $E = \sum e^2$. The autocorrelation method and Levinson-Durbin recursion algorithm [89] are used for minimizing the MSE so that the best possible sets of predictor coefficients are produced. Finally the LPC coefficients are calculated using the recursive LPC to cepstrum conversion routine according to Equation 2.6 below.

$$c_n = a_n + \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k} \tag{2.6}$$

2.1.6 Filterbank feature extraction

Filterbank feature extraction is implemented by passing a segment of speech signal through a sequence of band-pass filters of certain bandwidths and center frequencies and taking the mean of the output amplitudes of each of the band-pass filters as extracted feature. This arrangement of band-pass filters is in a sequence that leads to the feature extraction approach termed filterbank approach. This is because the human ear distinguishes frequencies non-linearly across the audio spectrum, a motivation on designing a front-end to operate in a similar non-linear manner developed.

These band-pass filters are often not linearly spaced on the frequency axis hence creating the similar manner of the human ear in distinguishing frequencies in an audio spectrum non-linearly. Experimental evidence has showed improvement in recognition performance under these filterbanks feature extraction [90]. The scales that are used to determine the width, centre frequency, and spacing of the band-pass filters in filterbank are Mel, Bark and the Erb scales.

One of the most popular filterbank based feature extraction method is the Mel-Frequency-Cepstral-Coefficients (MFCC), which is currently the most widely used in speech recognition systems. The filterbank it implements is made up of a sequence of

triangular bandpass filters whose widths and center frequencies are equally spaced on the Mel-scale defined as,

$$\mathbf{Mel}(f) = 2595 \log_{10}(1 + f/700) \quad (2.7)\ [3]$$

MFCC's band-pass filters are often triangular arranged such that they are nearly-linearly spaced at frequencies below 1000Hz and logarithmically spaced at frequencies beyond 1000Hz.

In implementing the MFCC, the magnitude of a spectrum of a window of speech is computed using Fast Fourier transform. At each frequency the magnitude of this spectrum is multiplied by the magnitude of the filter at the corresponding frequency. All the products from each bandpass filter on the filterbank are summed and used as extracted features. Similarly, these features can be completed with the LPC coefficients. Filterbanks outputs are highly correlated which makes the need for cepstral transformation almost inevitable to produce less correlated features. Given filterbank coefficients, cepstral coefficients can be computed using the Discrete Cosine Transform (DCT) as shown by the Equation 2.8 below,

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \log(m_j) \cos\left(\frac{\pi i}{N}(j - 0.5)\right) \quad (2.8)$$

where $\{m_j\}$ are output coefficients of band-pass filter on the filterbank and N are the number of filters used on the filterbanks.

2.1.7 Auditory Modeling feature extraction

The fact that humans are able to discriminate between noise and speech in noisy environment brought inspiration behind the development of feature extraction that model human auditory system (auditory modeling). If the human auditory system can be

successfully modeled and the models applied to speech recognition, performance in noisy environment would be improved. As well the amount of knowledge available about the human auditory system motivated this approach of feature extraction. Much of this knowledge only concerns the periphery of the auditory system i.e. as far as the sending of information by the nerve fibers to the brain. The interpretation of these signals in the brain is not certainly known.

One of the most popular auditory based feature extraction methods is a Perceptual Linear Prediction (PLP) [94]. The PLP analysis is to compute an auditory spectrum from which the LPC model is computed. This auditory spectrum is computed by convolving the power spectrum of a given segment of speech with some form of a critical band-masking pattern. The critical band spectrum (or auditory) is then pre-emphasized using a simulated equal-loudness curve and compressed using a cubic non-linearity, which is a simulation of the intensity loudness power law. From the resulting auditory spectrum a solution of a LPC model is computed. The LPC coefficients are taken as an extracted feature for the given segment of the speech. Cepstral coefficients can and are often computed from these features. The PLP can be viewed as combination of the LPC based and auditory modeling feature extraction principles.

Another feature extraction method that has been used recently in a number of research systems is known as the Mel-Frequency Perceptual Linear Prediction (MF-PLP) [99, 100]. As the name suggests, it combines the use of the Mel-filterbanks with PLP based principle. In this feature extraction method, Mel-filterbank coefficients are computed from the power spectrum of a given frame of speech. These coefficients are then weighted using an equal loudness curve and compressed by taking their cubic root. This results in an auditory spectrum as in the PLP case from which the LPC model coefficients are computed. These coefficients are taken as an extracted feature from which "cepstra" are computed. MF-PLP is simply the PLP feature extraction method applied on mel-filterbank coefficients rather than on the power spectrum of a given speech frame as it is in the normal PLP computation case.

Cambridge University Hidden Markov Toolkit (CU-HTK) group found MF-PLP to be more robust to mismatches in training and testing conditions than conventional methods like the MFCC and PLP [95]. The CU-HTK team has since used this front-end in virtually all their systems (see [96, 97]). These systems are large vocabulary continuous speech recognition systems evaluated on telephone conversational systems [98].

2.1.8 Other Components of Feature Vectors

There are other components that are included in feature vectors for the purpose of improving robustness of speech recognition systems. Rather than extracted features of speech, these measures are energy measures of given speech segments and time derivatives.

Energy

The inclusion of the energy measure in the parameter vector in speech recognition is commonly used. For the segment of speech N samples long this energy may be computed as [3],

$$E = \log \sum_{n=1}^{N} (w(n)s(n))^2 \tag{2.9}$$

where $w(n)$ is a windowing function. This energy may be computed without windowing. The log is often taken to simulate the logarithmic response of the human auditory system [90].

Time derivatives

Addition of time derivatives to parameter vectors has proved to enhance robustness of recognition systems and is standard across all recognition systems. There are three approximations of these derivatives. Their formulae with normalization factors dropped are [101],

$$\frac{d}{dt}s(n) \approx s(n) - s(n-1) \qquad (2.10)\ [101]$$

$$\frac{d}{dt}s(n) \approx s(n+1) - s(n) \qquad (2.11)\ [101]$$

$$\frac{d}{dt}s(n) \approx \sum_{m=-N_d}^{N_d} ms(n+m) \qquad (2.12)\ [101]$$

Equation 2.12 which is commonly used can be further expressed as,

$$\frac{d}{dt}s(n) \approx \sum_{m=1_d}^{N_d} ms(n+m) - s(n-m) \qquad (2.13)$$

First order derivatives are computed using one of the above stated approximations. Sometimes second order derivatives are computed from the first order derivatives using the same approximations. These derivatives are usually normalized using Equation 2.13 as follows,

$$\frac{d}{dt}s(n) \approx \frac{\sum_{m=1_d}^{N_d} ms(n+m) - s(n-m)}{2\sum_{m=1}^{N_d} m^2} \qquad (2.14)$$

2.2 Back-Ends: Classification and Pattern Matching

There are two processes that take place in the back-end of a recognition system depending on the mode of operation of the system. These are classified during training and acoustic pattern matching during recognition. The following are some of the well-

known pattern classifier and signal modeling schemes being used in speech recognition; Artificial Neural Networks (ANN) and Hidden Markov Models (HMM).

Artificial Neural Networks are an attempt to model the biological nervous system as a trainable set of mathematical models. In speech recognition they are used as a classification system. Different types of NNs have been used in speech recognition such as Time Delay Neural Network (TDNNs) and Self Organizing Maps (SOM) [102]. One problem that has hindered successful use of NNs in speech recognition is their inability to deal with time warping which renders them not useful when large time-spans are integrated in systems [102]. It is for this reason that NNs are often used in hybrid systems in combination with other classifiers like HMMs. In these hybrid systems they are put after the classifiers to manipulate their outputs.

Hidden Markov Models are the current modeling technique of choice in speech recognition. The theory of Hidden Markov Models (HMM) was published in the late 1960s and early 1970s by Baum and his colleagues [80, 81, 82, 83, 84]. It was first implemented in speech recognition by Baker [85] at CMU (Carnegie Mellon University), and by Jelinek and his colleagues at IBM in the 1970s [86]. Since speech recognition systems built in this study are HMM based, HMMs theory and applications are described in more details in the next subsections.

2.2.1 Hidden Markov Models (HMM) Definition

HMMs have been derived from Markov Model, which is intended to model certain types of processes. For a finite (non-hidden) set of states, S = $\{ s_1,...,s_N \}$, which consists of N unique states, the process being modeled is in exactly one state from S in any time instance. Moreover, the process enjoys the Markov property; the history that led to the current state is irrelevant to the future behavior of the process [87]. All that matters to future behavior is the current state. At each time, the process transitions to the next state (which may be the same as the previous state), based upon a transition probability

distribution, a_{ij}, depends only on the previous state. Here a_{ij} is the probability that, when in state i the process will transition next to state j.

This set of probabilities obey the following statistical constraint,

$$\sum_{j \in S} a_{ij} = 1 \qquad \forall i \in S \tag{2.15}$$

This kind of a Markov chain may be referred to as an observable Markov model because its process produces a sequence of states each of which is associated with a particular observable event from a defined observation set. This implies that every state sequence has a corresponding certain observation sequence.

A finite Hidden Markov Model (HMM) are a special case of Markov chain in which the observed event, when a state is reached, is not a definite element of a set of all possible observations but is determined by a probabilistic function of the state. This is to say that there is no one-to-one mapping between a state and an observed event. These observations, $\Theta = \{\theta_1, ..., \theta_M\}$ which consist of M unique states, can only be produced through an underlying set of stochastic processes. A distribution of such probability functions is defined as $b_i(k)$ which is the probability of observing θ_k when the process is in state i.

HMM observes the properties of Markov model and the fact that the probability of producing some observation in each state is 1:

$$\sum_{k \in \Theta} b_i(k) = 1 \qquad \forall i \in S \tag{2.16}$$

Before the HMM process (transition from one state to another for an allocated time units) starts, initial parameters are required. This initial state is defined as $B = B_i$, representing the probability that the process begins in state i where

$$\sum_{i \in S} B_i = 1 \tag{2.17}$$

We use λ to refer to a particular HMM model, i.e.,

$$\lambda = (S, \Theta, a, b, B) \tag{2.18}$$

HMMs pose three problems that need to be solved in order to apply them which are discussed in the next section. In this study the sequence of observations is referred to as $O = < O_1, ..., O_T >$ where T is the length of the observation sequence and where each $O_t \in \Theta$. The state sequence that produced O is referred to as $I = < I_1, ..., I_T >$.

2.2.2 Three Problems for HMMS

The three central problems that arise when working with HMMs [87].

- **Problem 1**, given O and λ, calculate $P(O \mid \lambda)$, i.e., the probability that this model, λ, will produce the observation sequence O. This is the *evaluation problem:* How likely is the model to produce the observation sequence?
- **Problem 2**, given O and λ, choose a state sequence $I = < I_1, ..., I_T >$ that is "most likely" to have produced O from λ. This is the *uncovering problem*: Which sequence of state transitions is most likely to have led to this sequence of observations?
- **Problem 3**, given O and λ, how can we adjust the model parameters *a, b*, and *B* to maximize $P(O \mid \lambda)$. This is the *training or learning problem*: How can we take a set of observation sequences and learn the best values for the model parameters *a, b*, and *B*?

Solving Problem 1: The Evaluation Problem

One method is to use the law of total probability and condition on all possible state sequences, *I*, of length *T*. This approach is computationally not feasible. It involves $(2T-1)*N^T$ multiplications and N^T-1 additions [87]. This makes it a poor method. The standard approach is to use the forward-backward procedure, which takes advantage of the Markov property. The path to a state is irrelevant to the future behavior of the process after being in that state. This algorithm is explained in the following subsection.

The Forward Algorithm

Consider the forward variable $\alpha_t(j)$ defined as:

$$\alpha_t(j) = P(O_1,...,O_t, I_t = j \mid \lambda), \tag{2.19}$$

Which is the probability of the process producing the first t observations and ending up in state j at time t. $P(O \mid \lambda)$ is easily calculated from α_T. To visualize, the chain rule is applied and yields $\alpha_t(j) = P(I_t = j \mid \lambda) \times P(O_1,...,O_t, I_t = j \mid \lambda)$. However, the law of total probability relates P as below,

$$P(O_1,...O_t \mid \lambda) = \sum_{j \in S} [P(I_t = j \mid \lambda) \times P(O_1,...,O_t, I_t = j \mid \lambda)] = \sum_{j \in S} \alpha_t(j) \tag{2.20}$$

which becomes

$$P(O \mid \lambda) = \sum_{j \in S} \alpha_T(j) \tag{2.21}$$

if the entire observation has to be considered. The initial value for this algorithm, $\alpha_1(i)$, is set to

$$\alpha_1(i) = B_i b_i(O_1) \tag{2.22}$$

Notice that calculating $P(O \mid \lambda)$ requires on the order of $N \times T$ calculations using α_T as compared to requiring an exponential number of calculations [87]. There is another method similar to the forward algorithm known as the backward algorithm. The algorithm is briefly explained in the following subsection.

The Backward Algorithm

The backward variable $\beta_t(j)$ is defined as,

$$\beta_t(j) = P(O_{t+1}, O_{t+2}, \ldots, O_T \mid I_t = j, \lambda) \tag{2.23}$$

This is the probability of the process being in state j at time t and of producing the observations from time $t + 1$ to T. The objective at this point is calculating the probability of being in state i at time t and masking a transition to state j at time $t + 1$ and producing the rest of the given observation sequence from O_{t+1} to O_T for all states j.

Such a probability is expressed as,

$$\beta_t(i) = \sum_{j \in S} a_{ij} \times b_j(O_{t+1}) \times \beta_{t+1}(j) \tag{2.24}$$

The initial value for this computation is set to,

$$\beta_T(i) = 1 \tag{2.25}$$

This algorithm yields the same results as those of the forward algorithm with the same computation efficiency. As will be seen, these two algorithms are together used to solve the training problem of HMMs.

Solving Problem 2: The Uncovering Problem

The criteria upon which to maximize our selection of I must be identified first. This is needed to maximize $P(I|O, \lambda)$ which is equivalent to maximizing $P(I,O|\lambda)$. Iterate through all possible such I and select one with the maximum value. This method is infeasible given the number of possible state sequences, therefore a more efficient method that takes advantage of the Markov property is ideal. The well-known solution to this problem is known as the Viterbi Algorithm [88].

To explain this algorithm we define the quantity $\delta_t(j)$, as the highest probability along a single path at time t that accounts for t observations from O and ends at state j. This is expressed by the following equation.

$$\delta_t(j) = \max_{I_1,\ldots,I_{t-1}} P(I_1,\ldots,I_{t-1}, I_t = j, O_1,\ldots,O_t \mid \lambda) \tag{2.26}$$

and hence meaning that,

$$\delta_t(j) = \max_i [\delta_{t-1}(i)a_{ij}b_j(O_t)] \tag{2.27}$$

To solve the problem, Equation 2.27 is computed recursively starting from the initial value,

$$\delta_1(j) = B_j b_j(O_1) \tag{2.28}$$

until the termination condition expressed by Equation 2.29 is reached,

$$P = \max_{i \in S}[\delta_T(i)] \tag{2.29}$$

Along the recursion process the optimal states are produced, therefore there is a need for an array to store these states. This array, $\psi_t(j)$ is defined as

$$\psi_t(j) = \arg\max_i [\delta_{t-1}(i)a_{ij}] \tag{2.30}$$

with the initial value set to,

$$\psi_1(j) = 0 \tag{2.31}$$

Finally the most likely state sequence is then found to be,

$$I_t = \psi_{t+1}(I_{t+1}) \tag{2.32}$$

in speech recognition systems this algorithm is used in the recognition process. The acoustic vector computed from the speech signal to be recognized serves as the given observation sequence. From the given HMM model network the most likely state sequence to have produced the given acoustic data is computed. The state sequence is then mapped into words using a dictionary.

Solving Problem 3: The Training Problem

The training problem attempts to adjust the parameters of the given model so as to maximize its probability of producing the given observation sequence, i.e. to maximize $P(O \mid \lambda)$. The method used to solve this problem is an iterative technique known as the Baum-Welch method [80, 81, 82, 83, 84]. It is named after L.E. Baum the developer of the method.

Training is particularly difficult since only a set of observation sequences that the process actually produced is given and the associated state transitions that occurred are hidden. Given these actual state transitions, the model adjustment would be simpler. But without those hidden variables, one must guess the state where the transition occurred.

If $\gamma_t(i)$ is summed up over t, what is obtained is the expected number of times that state i is visited, or equivalently, the number of transitions made from state i, if the last time point is excluded. Thus, $\sum_{t=1}^{T-1}\gamma_t(i)$ i.e the expected number of transitions made from state i and $\sum_{t=1}^{T-1}\xi_t(i,j)$ which is the expected number of transitions made from state i to state j. expressed in terms of these summations the re-estimation formulae of HMM parameters are,

$$\overline{B_i} = \gamma_1(i) \quad \text{for } i \in S \tag{2.33}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{2.34}$$

$$\overline{b_i(k)} = \frac{\sum_{t=1}^{T}\left[\gamma_t(i)\times\begin{Bmatrix}1 \text{ if } O_t = k\\ 0 \text{ otherwise}\end{Bmatrix}\right]}{\sum_{t=1}^{T}\gamma_t(i)} \tag{2.35}$$

The numerator of Equation 2.35 only includes t such that (i.e., *s.t.*) $O_t = k$, where k is the observation being examined. The training procedure, using this algorithm, can be summarized in the following steps:

1. Initialize the HMM model $\lambda = (S, \Theta, a, b, B)$
2. Compute $\xi_t(i,j)$, $\alpha_t(j)$ and $\beta_{t+1}(j)$ and $\gamma_t(i)$
3. Estimate $\overline{\lambda}$ from $\xi_t(i,j)$ and $\gamma_t(i)$
4. Replace λ with $\overline{\lambda}$
5. If not converged return to 2

It can be shown that unless $\overline{\lambda} = \lambda$ (re-estimation iteration termination condition) $P(O|\overline{\lambda}) > P(O|\lambda)$. In speech recognition, these formulae are used to re-estimate (iteratively update and improve) parameters of the given model from the given training data.

2.2.3 Continuous Observation Density HMMs

The discussion on HMMs up to the last section refers to the case in which the HMMs are used with discrete observations. However, for many applications the observation set is not discrete but continuous if not made up of vectors. Continuous speech recognition is one such application of HMMs in which the observation set is not discrete. Hence it would be advantageous to be able to use HMMs with continuous observation densities to model continuous signal representations directly.

In this type of HMMs the discrete observation density $b_j(k)$ is replaced by a continuous observation density $b_j(\boldsymbol{x})$ where $\boldsymbol{x}$ is a vector. There is usually more than one of these observation densities per state. The observation density of a finite number of mixtures is therefore expressed as,

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{M} c_{jk} \chi(\boldsymbol{x}, \mu_{jk}, U_{jk}) \tag{2.36}$$

Where c_{jk} the observation density coefficient for the kth observation density is in state j. The term χ can be any log-concave or elliptically symmetric distribution. In this case it is assumed to be a Gaussian with mean vector μ_{jk} and covariance matrix U_{jk} for the kth mixture (observation density) component in state j. the mixture coefficients must satisfy the following stochastic constraints,

$$\sum_{k=1}^{M} c_{jk} = 1 \tag{2.37}$$

$$\text{where, } c_{jk} \geq 0$$

In speech recognition the observation vectors are made up of a feature extracted from speech signals in the front-end (signal processing end) of a recognizer. There are many feature extraction methods and HMMs can be used with any of them.

2.2.4 HMMs in Speech Recognition

An HMM can be used to model a specific unit of speech. This specific unit of speech can be a phoneme, a word, or a complete sentence or paragraph. In isolated speech recognition or small-vocabulary system, HMMs are usually used to model words while in continuous speech recognition or large-vocabulary systems, HMMs tend to be used to model sub-words units such as phonemes. Since all the recognition systems built in this study are continuous speech recognition systems, hence we focus on the continuous speech recognition case. The process of building a continuous speech recognition system often starts with the construction of a recognition network followed by the training of the HMM models in the network.

A recognition network can be viewed at three levels once it has been constructed. The highest level in this network is the word level and is mapped to the next level which is the phones level using the dictionary. In the dictionary, all words in the training data set are listed together with their phonemic composition. Each of the phones in the phones level has an HMM model and all of these models are concatenated to make up the lowest level of the network which is the HMM model network. There are nodes on such networks that are either HMM model instances or word-ends.

Depending on the vocabulary size, the networks can be constructed differently. For small vocabulary size, these networks are usually constructed according to a defined *task grammar, which* defines all allowable word sequence. However, for medium vocabulary and large vocabulary (e.g. desktop dictionary systems) it is not possible to define such a grammar. In such cases a *word loop* is used which puts all words contained in the system's vocabulary into a loop such that any word in the vocabulary can follow any other word. These word loops are usually augmented by a statistical language model (e.g. bigram). In this study, all the recognition systems built are medium vocabulary and their word loops have been augmented with a bigram language model. Once successfully constructed, the models in the recognition network are trained.

The training procedure uses the feature vectors from each of the given utterances in the training data set, which were extracted from the front-end or the signal-processing end of the recognizer. These features are considered as observation vectors in the training of HMMs. The utterances used for training must be segmented and properly labeled according to the phone set used. Therefore to train all HMM models of phones appearing in the labeled utterances can be made possible.

The recognition problem is that of calculating the most likely word sequence from the given sequence of features using the recognition network. Firstly, the utterance to be recognized must be processed to extract the features that serve as observations. These are then used to compute the most likely state sequence to have produced them from the recognition network. Thereafter this state sequence is mapped to the phones sequence then to the word sequence.

Recognition is done by searching for the best path in the recognition network and the one with the highest probability is singled out as the recognition result. This search is done by using the Viterbi decoding algorithm. This algorithm is used to find the optimal state sequence rather than the optimal word sequence. In the experiments performed this basic algorithm is implemented within a search algorithm known as the *Token Passing* algorithm [3]. A token represents a partial path on the network from time 0 to time t. In each time step, transitions are made continually along the connected states, stopping when the last state on the HMM model is reached. As these transitions are made along potential paths the history of the nodes passed is recorded. At the end of the transitions the back-trace of the transitions is output as the recognition results.

2.3 Summary

In this chapter, the concepts in speech recognition technology have been discussed including both the front-end and the back-end. In the next chapter, principle of speech coding with more emphasis on the GSM speech codec is discussed.

Chapter 3

Speech Coding in GSM

This chapter covers the speech coding schemes employed in the Global System for Mobile Communication (GSM) channels. Since this study seeks to improve robustness of GSM based speech recognizer, this chapter is concentrated on GSM speech coding algorithms.

Speech coding and compression is an important element in efficient transmission of speech over digital wireless systems [15, 16]. The objective in speech coding is to represent speech with a minimum number of bits while maintaining its perceptual quality. Speech compression has been central to the technologies of robust long-distance communication, high-quality speech storage, and message encryption [15]. Speech coding and compression is the process of sampling analogue speech signals and then efficiently compressing them into digital bit streams. A decoder then receives the bit stream and decompresses it back into a speech signal. Speech coding algorithms or speech coders are therefore composed of two parts, an encoder and a decoder. The parts of a speech coder are illustrated in figure 3.1 which also shows transmission of digital speech from the source to the receiver.

In telecommunications, speech coding allows for a more efficient use of the available bandwidth in the sense that the lower the bit rates the more communication channels can

be fitted within a certain limited bandwidth. Since low bit rates means low bit counts of information, speech coding for more efficient use of available storage space.

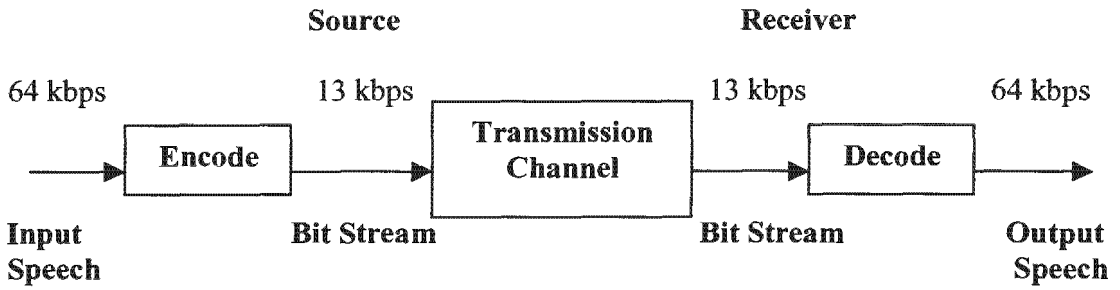


Figure 3.1: Block diagram of the transmission of digital speech

3.1 Coding algorithms

There are three broad classes of speech coders, waveform coders, vocoder, and hybrid coders. These are discussed below.

3.1.1 Waveform coders

Waveform coders are signal source independent. This means they can simply digitize a speech signal without considering its source or how it was produced and they work with non-speech signals. They attempt to produce a signal that is as close as possible to the original at the decoder. The resulting data rates in waveform coders vary from 16 kbps to 32 kbps. These data rates ensure that the quality of speech that is produced is high. Waveform coders are of low complexity and are inexpensive. However, their main disadvantage is their bandwidth inefficiency due to their high bit rates [16].

The simplest form of wave coding is Pulse Code Modulation (PCM). In PCM, speech is sampled at the Nyquist rate. A binary number represents each sample obtained. The samples are logarithmically compressed and binary encoded. At the receiver, binary

decoding occurs followed by logarithmic expansion to recover the speech [16]. Other examples of such coders are Differential Pulse Code Modulation (DPCM), Delta Modulation (DM) and Embedded Delta Modulation (EDM).

3.1.2 Vocoders

Unlike waveform coders, vocoders are designed specifically for speech and would not work well with other types of signals [16]. They are known as Source coders. They take advantage of redundancy in speech signals to compress the signal by modeling its source (i.e. the vocal tract and the vocal cord vibrations) using digital filter. The speech synthesis section of vocoders models the speech generation process via a basic model known as the source-filter model. The model consists of an excitation signal (source), which represents the air that is modulated by the vocal cords and a filter to characterize the vocal tract [16].

Parameters for the filter of this model are extracted from a speech waveform at the encoder. These parameters, rather than the speech waveform, are transmitted as a stream of bits. The decoder then converts the bit stream back into model parameters. These are then used to produce a speech signal, which is perceptually close to the original. Vocoders represent speech with fewer parameters than waveform coders. By doing so, they are able to achieve higher speech compression ratios than waveform coders. Although they can be operated at lower data rates, this produces poorer speech quality. Generally, the lower the bit rate the lower the speech quality. It is for this reason that previous research has shown that the lower the bit rates of speech the lower the recognition performance on such speech [17].

A popular example of a vocoder is the Linear Predictive Coding (LPC). Since the speech coders used in GSM networks are based on this LPC, it will be discussed in this study.

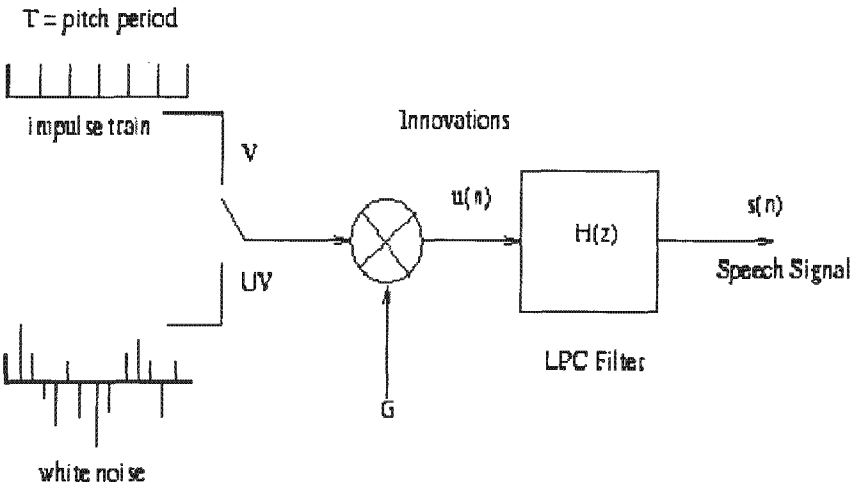


Figure 3.2: Speech production model used in LPC

LPC-Linear Predictive Coding

Most of the current low bit coders are LPC-based. The vocoder analyzes a speech signal and transmits parameters derived from this analysis. The signal is then synthesized at the receiver using these parameters. LPC is able to conserve bandwidth by parameterising the speech signal. It is therefore one of the most powerful speech analysis techniques [16]. The codec uses a bit rate of 2.4 kbps and although it produces accurate estimates of speech parameters, it results in an intelligible artificial sounding speech. LPC models the vocal tract as a time varying filter. The model assumes that speech is produced as a result of exciting the LPC filter with a source signal. The speech production model used in LPC is shown in figure 3.2 [16].

The figure 3.2 shows that the excitation signals for the filter can be either a "train of impulses" or "white noise". An impulse train represents voiced sounds while unvoiced sounds are represented by white noise.

The vocoder is composed of two parts: an encoder and decoder. At the encoder, two processes occur. First, the obtaining LPC filter parameters from speech samples and

second, determining the ideal excitation sequence for the LPC filter. The filter parameters and excitation sequence are transmitted to the receiver instead of the speech signal itself. At the decoder, the excitation signal is then passed through the synthesis filter to reconstruct the speech.

To calculate the filter parameters, Hamming window is used with windowed frames of 20ms. Thereafter, linear prediction that involves predicting the current sample of the windowed speech using a linear combination of past samples. This is done by forming the prediction error, *e(n)* between the sample, *s(n)* and its prediction, $\hat{s}(n)$ as shown in the equation 3.1 below.

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad (3.1)\ [16]$$

a_k represents unknown predictor parameters or coefficients, p is the number of predictor coefficients used called predictor order. The predictor coefficients can be found by computing and minimizing the mean square error, E, between the original signal and an estimated speech signal over a finite duration. Computation of the mean square error, E, is shown in the equation 3.2 below.

$$E = \sum_{n} e^2(n) = \sum \{s(n) - \sum_{k=1}^{p} a_k s(n-k)\}^2 \qquad (3.2)\ [16]$$

To minimize the error, its partial derivative is set to zero as expressed in the equation 3.3 below.

$$\frac{\partial E}{\partial a_l} = 0 \qquad (3.3)\ [16]$$

The predictor coefficients can then be computed using processes known as auto-variance or auto-correlation for every 20 ms speech frame. These processes are clearly discussed and documented [15], [16].

As discussed previously, the predictor coefficients that are formed are used to form the filter, H(z). This filter will then be used to synthesize the speech signal. The transfer function for the time-varying filter is represented by equation 3.4.

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-i}} \qquad (3.4)\ [16]$$

Where G is the gain of the filter and a_k represents the predictor coefficients.

3.3.3 Hybrid coders

Hybrid coders are a combination of waveform coders and vocoders. This combination is able to produces good quality sound at low to medium bit rates. This is why hybrid coders are utilized in digital communication systems. Examples of hybrid coders are Regular Pulse Excited- RPE (used in GSM full rate speech coders) and Code Excited Linear Prediction (CELP).

The classes of speech quality coding algorithms vary with the bit rates of the various coders. The figure 3.3 obtained from [15], illustrates the relationship between the speech quality of the coding algorithms and their bit rates. It shows that waveform coders generally have higher data rates than the other two coders. This illustration shows that vocoders produce the poorest speech quality.

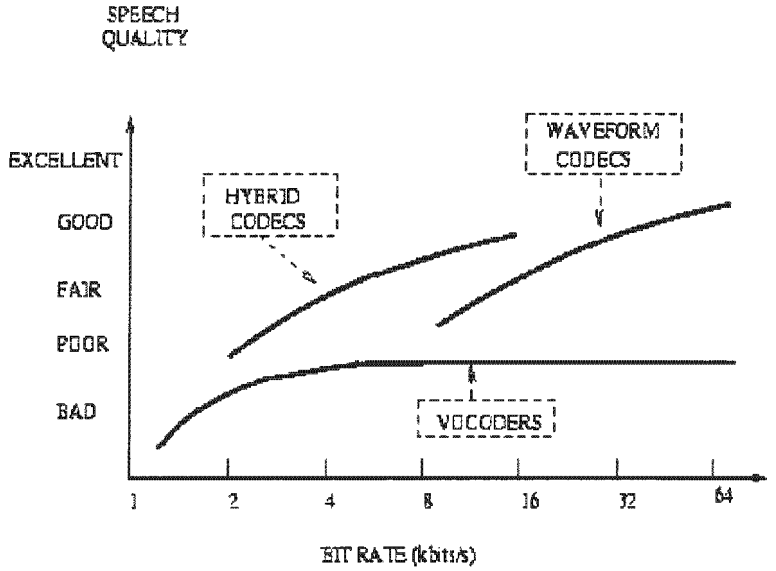


Figure 3.3: Speech quality versus bit rates for the three classes of speech coding algorithms [15]

3.2 GSM- Global System for Mobile Communication

GSM is the most widely used digital wireless technology in the world [19]. Currently, there are over 1.05 billion GSM users globally and 18 million users in South Africa [19, 20]. A summary of GSM milestones is given in a self summarized table 3.1 [19].

3.2.1 Speech Compression in GSM

There are four types of GSM speech-coding algorithms [16]. These are, full-rate (FR), half-rate (HR), enhanced full-rate (EFR) and adaptive Multi-Rate (AMR)- this is a new technology which is discussed further in [21].

TABLE 3.1

GSM MILESTONES

1981	Introduction of analogue cellular and discussion on digital cellular and creating a pan-European standard to combat the incompatibility of different analogue systems
1982	A study group called GSM-Groupe Special Mobile is formed to develop a possible pan European digital cellular standard
1987	18 countries sign a MOU (Memorandum of Understanding) stating that by 1991 a working digital cellular system would be in place
1991	UK, France, Germany, Italy introduce digital cellular services
1992	Motorola start first commercial GSM system
1993	Groupe Special Mobile changes its name to Global System for Mobile Communication

Full Rate (FR) Speech Coder

The FR coder was standardized in 1987. This coder belongs to the class of Regular Pulse Excitation – Long Term Prediction- linear predictive (RPE-LTP) coders [22, 23]. The full rate coder consists of an encoder and decoder. A speech applied to the encoder, is split into frames that are 20 ms long. Each frame, which consists of 160 speech samples, is encoded to become 260 bits of data. Since 260 bits are encoded every 20 ms, the overall bit rate of the coder is 13 kbps. The decoder maps the encoded blocks of 260 bits to output blocks of 160 reconstructed speech samples [23]. The GSM full rate channel supports 22.8 kbps. Thus, the additional 9.8 kbps are then used for error protection.

Half Rate (HR) Speech Coder

The HR coder standard was developed with lower bit rate to cope with and accommodate the increasing number of subscribers. This is a 5.6 kbps Vector Sum Excited Linear Prediction (VSELP) coder [24]. In order to double the capacity of the GSM cellular system, the half rate channel supports 11.4 kbps. Therefore, 5.8 kbps are used for error

protection. The measured output speech quality for the HR coder is comparable to the quality of FR coder in all tested conditions [25], except for tandem and background noise condition.

Enhanced Full Rate (EFR) Speech Coder

The EFR coder is a 12.2 kbps Algebraic Code Excited Linear Prediction (ACELP) coder. Although similar in principle to the RPE-LTP, there are differences. One of these differences is that the EFR uses a 10t order linear predictive filter.

3.2.2 Multiple Access in GSM

The major issue with the use of a wireless system is that users share a common communication channel. This brings conflicts when many users wish to transmit data or use resources at the same time. Multiple access protocol rules are required to regulate how a communication channel capacity is allocated to the users. Three multiple access protocols are time division multiple access (TDMA), frequency division multiple access (FDMA) and code division multiple access (CDMA) [16]. GSM uses TDMA to transmit and recover data that has been coded by the RPE-LTP vocoder [16].

The GSM standard was decided after running a number of comparative experiments to determine the best among different digital systems, which were then proposed. These speech coders were compared in terms of speech quality, robustness to channel errors, processing delays and computational complexity; the RPE-LTP was found to be the best and therefore selected as the coding scheme.

3.2.3 Advantages and Disadvantages of GSM

This section highlights some advantages and disadvantages that are associated with utilizing GSM networks.

Advantages

- Offers extensive coverage of 665 networks in 179 countries that provides better capacity for the worldwide roaming [19].
- Good voice quality.
- Has a wide international user base and offers a mature network that could easily integrate new technologies [21].

Disadvantages

The widespread growth of the wireless industry requires a system that can accommodate more users. GSM systems have a limited capacity due to the TDMA [19]. Unlike CDMA, which allows full utilization of system resources such as bandwidth and time to accommodate the increasing number of users, TDMA cannot easily allow for more users on the GSM network.

3.2.4 Effects of GSM speech coding on Recognition Performance

Huerta [25] proved that the short-term residual of the RPE-LTP coder used in GSM produces a significant level of quantization distortion that in turn affects GSM speech recognition performance. This distortion affects different classes of phones differently. From Huerta's results in experimenting with these different classes of phones it was reported that the most severely affected class is mostly made up of nasals. Lilly and Paliwal [17] showed and concluded that reduction in bit rate of speech due to speech coding affects recognition performance.

3.3 Summary

In this chapter different speech coding schemes have been discussed. The three classes of speech coding algorithms namely waveform, vocoder and hybrid were then listed and compared. A description of the LPC vocoder on which speech coding algorithms used in GSM networks is based followed. Then a discussion of GSM, the globally accepted standard for mobile communication was presented. Four speech-coding algorithms utilized in GSM were mentioned. Lastly the effects of RPE-LTP on speech recognition

performance are outlined. In the next chapter the experimental setup and baseline results in this study are discussed.

Part 2

Theory of techniques used and experimental work

Chapter 4

Experimental Setup and Base-line Results

In the first three chapters we introduced the background work covering the theory of speech recognition and the speech coding schemes on GSM. From this chapter onwards we introduce the theory of techniques used and experimental work on building the recognition systems investigated in this study.

This chapter describes the work done in developing the recognition systems using HTK tools and the baseline recognition results for this study. Even though speech recognition systems differ with tasks for which they are developed, their development is generally the same.

The development process usually starts off with the construction of what is known as a *base-line recognizer*. The base-line performance is measured in percentage Word Error Rate (% WER) or the percentage word recognition accuracy it produces on a given task. From this point on, experiments are conducted to improve the performance of this base-line until the optimum performance is obtained on the given task (See, for an example, [7, 14]). This might involve fine-tuning certain parameters of the base-line system. The next section starts with details of the general design for all recognition systems in this study.

4.1 General System Design

In this study, all the recognition systems were developed using Hidden Markov Model Toolkit version 3.2.1 (HTK v3.2.1). Listed below are the aspects of the recognition systems, which were applied to each of the systems implemented.

- The systems are phone-based continuous speech recognition systems with a total of 49 phones.
- The systems are HMM-based recognition systems, the HMM models for each phone are simple left-right with 5 states in total, 3 of which are emitting with continuous Gaussian mixture density. The number of HMM mixtures were incremented to achieve the best performance.
- TIMIT, NTIMIT and GSM-TIMIT were the databases used to evaluate the recognition performance.
- The TIMIT dictionary [1] was used with 61 phones set mapped to CMU's 48 (including silence) phones set listed in [2]. With the addition of the short-pause (sp) the number of phones in total was 49.
- The systems are tied-state tri-phones
- For all the front-ends, feature extraction was performed on a hamming window of 25ms in every 10ms, which is applied on each frame of speech with a pre-emphasis coefficient of 0.97. 12 static features plus an energy term were computed and attached with both delta and acceleration coefficient for all features to complete a 39-feature vector size.

4.2 Speech databases

Before any experimental work could be carried out, a database of speech samples had to be chosen. Three databases namely TIMIT, NTIMIT and GSM-TIMIT were considered. In this section we provide detailed description of the phonetically-labelled TIMIT, NTIMIT and GSM-TIMIT databases that are used to evaluate the effect of Telecommunication channels on the HTK recognition system.

4.2.1 TIMIT

The TIMIT database was designed to provide speech data for the acquisition of acoustic-phonetic knowledge and for the development and evaluation of automatic speech recognition systems. It is made up of 630 speakers from 8 major dialect regions of the United States, each saying 10 sentences, making a total of 6300 sentences. The speech material has been subdivided into portions for training and testing. The criterion for the subdivision is as follows:

- Roughly 20 to 30 percent of the database should be used for testing purposes, leaving the remaining 70 to 80 percent for training.
- No speaker should appear in both the training and testing portions.
- All the dialects regions should be represented in both subsets, with at least 1 male and 1 female speaker from each dialect.
- The overlap of text material is minimal; if possible no text should be identical.
- All the phonemes should be covered in the test material; preferably each phone should occur multiple times in different context.

The more detailed descriptions of the subdivisions are available in the file [1].

The TIMIT database in this study was used for clean speech test results and training set. The clean speech recognition system evaluations were done using the core test set [1]. The TIMIT dictionary [1] was used with its phones set mapped to CMU's 48 (including) phones set listed in [2].

4.2.2 NTIMIT

The NTIMIT database is database of the TIMIT that has been passed through the telephone network in the USA. The database was not simulated nor was it filtered to fit the channel, but it was passed over the telephone network in the raw form. It is

phonetically equivalent to the TIMIT database. In this study this database was used for further experiments on the robustness of the proposed methods.

4.2.3 GSM-TIMIT

The CTIMIT was the other database that was available for use. It is a cellular version of the TIMIT database. This database could not be used in this study because first it was passed through the American analog cellular network and not through the international digital GSM network, and second it is not complete. This limited its usefulness to us.

Due to the unavailability of GSM data, recording of GSM speech files were made over the Vodacom network (South African Cellular Company which uses Enhanced full rate GSM speech codec [114]) for testing. The recording was done similarly to the NTIMIT recording. Two Mobile Stations (cellular phones) were used, one as a transmitter end and the other as a receiver end. The transmitter end station was used to transmit speech signals from TIMIT database core test set [1] over the GSM network to the receiver end where it was recorded. The two ends were two computers appropriately networked over the IP network for synchrony purposes. The speech were played and recorded at 16 kHz. For the purpose of experiments, they were down sampled to 8 kHz (producing an effective bandwidth of 4 kHz). All experiments, both training and testing were performed with 8 kHz sampled speech files.

4.3 HTK, a Hidden Markov Model Toolkit

This section is a description of the modules of HTK, which were used in this study. The Cambridge University Engineering Department and Entropic Research Labs developed HTK. Different versions with improvements and new features have been available. Version 3.2.1 was used for all results reported in this study. HTK has software library modules and user-level tools for speech analysis, model training, Viterbi recognition, results analysis as well as interactive speech analysis. HTK is not a ready-made speech

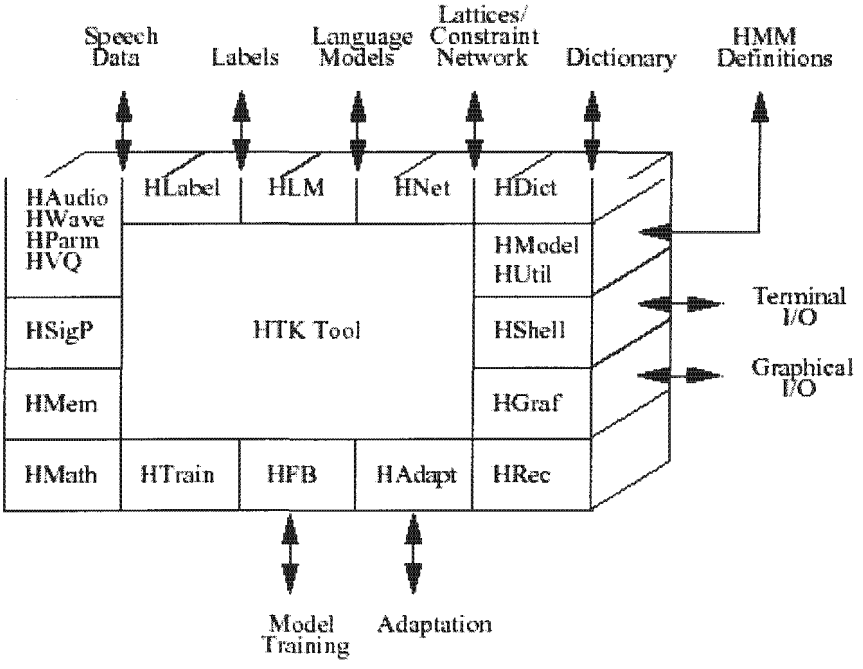


Figure 4.1: HTK software structure (from [1])

recognition system but a toolkit made up of tools coded in C programming language that one needs to construct a complete speech recognition system. This means one does not have to write programs for a speech recognition system from scratch but needs to know how to use these tools properly to develop recognition engines. HTK can be used to build systems that recognize isolated words, connected words, or continuous speech.

This toolkit was chosen in this study because it can create a basic speech recognition system from scratch and since the source code is available, it opens the window of opportunity for modification. Figure 4.1 illustrates the software structure of a typical HTK tool. The overall structure with processing stages is shown in figure 4.2. The command names which all start with "H" are in square boxes. There are 4 main phases as shown in the figure: data preparation, training, testing and analysis. In this section these phases are briefly discussed.

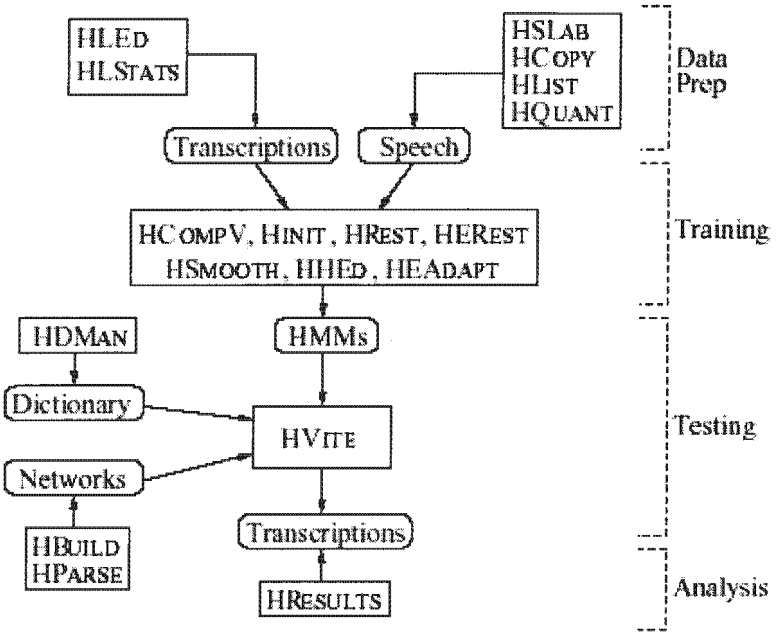


Figure 4.2: HTK processing stages (from [1])

4.3.1 Data Preparation

Before model training and recognition can be done, the data needs to be processed both for use with HTK and for use in model training and recognition. This means that the speech waveforms need to be converted into the appropriate parametric form of feature vectors, a dictionary of words and their phones mapping (pronunciations) needs to be created, and transcriptions must exist for all training and test speech. This must all be in HTK format [3]. In this section, descriptions on how to prepare a dictionary, transcription files, and encoded speech are briefly discussed.

Dictionary Preparation

The dictionary used is the TIMIT database dictionary [1]. As it is, it is not compatible with HTK tools, so the format of the database was altered to an HTK compatible format. Given a phoneme set, there should be some phoneme mapping for every word in the training and test set vocabulary. HTK allows for multiple pronunciation as well as output

symbols that do not necessarily match the word that is useful for sentence begin and end markers. The HTK tool, **HDMan** can be used to edit and merge differing source dictionaries to form a new dictionary. For instance, the British English BEEP pronunciation dictionary can be modified to form a new dictionary by adopting its phone set without modification, except that the stress marks will be removed and a short-pause (sp) will be added to the end of every pronunciation. HDMan can as well be used to convert a monophone dictionary to one containing biphones and triphones.

A sample format to show how the dictionary is like is shown in Figure 4.3. It shows multiple pronunciation, as well as optional outputs, shown between square brackets.

```
A               ax sp
A               ey sp
CALL            k ao l sp
AND             ax d d sp
AND             ae n d sp
EIGHT           ey t sp
EIGHTEEN        ey t iy n
SENT-END  []    sil
SENT-START []   sil
ZERO            z ia r ow sp
TO              t ax sp
TO              t uw sp
```

Figure 4.3: Sample dictionary format, including alternative pronunciations

Creating Transcription Files

Assuming that there are word transcriptions for the speech in some format, there are scripts available to convert them into HTK format files. In HTK, every file of training data must have an associated phone level transcription. The HTK tool **HLEd** is a simple editor for manipulating label files and can read in label files that are in TIMIT, SCRIBE, ESPS or HTK format. The HTK label format contains the actual label optionally

preceded by start and end time, and optionally followed by a match score. Typically the name of the label file will be the same name as the corresponding speech file but with a different extension. When very large numbers of files are being processed, label files access can be greatly facilitated by using master label files (MLFs) which may be regarded as index files holding pointers to the actual label files. An edit command script used in conjunction with HLEd can produce such changes as conversion from words to phones, or monophones to triphones, renaming or deletion of phonemes if moving to a different set of phoneme labels, or to manipulate the labels in an MLF [4].

Speech Processing

The final stage of data preparation is to convert the waveforms into a sequence of feature vectors. The HTK tool **HCopy** is used for that purpose as well as copying waveform files, or portions of the files, of different formats (NIST, Esignal, TIMIT) to HTK format. To do so, a configuration file is needed which specifies the parameters of the signal processing. Reasonable settings for these are as shown in Table 4.1. In brief, the example on Table 4.1 specifies that target parameters are to be MFCC using E (a normalized log energy coefficient), the frame period is 10msec, the output should be saved in compressed format. The FFT should use Hamming window and apply preemphasis of 0.97. The filter banks should have 26 channels, 12 MFCC coefficients should be outputed and energy normalization should be performed. For further options and equations on implementing the different parameter see [3].

4.3.2 Model Training

The second step of system building is to define the parameters that best describe their distribution by training both language and acoustic models. Together the language and acoustic models model aspects of the speech. The language model defines the probability of a sequence of words, while the set of acoustic models describes the probability of a sequence of phonemes given the sequence of words and together can be used to find the most likely word string [4]. In this section we discuss how models are trained using HTK.

TABLE 4.1

CONFIGURATION FILE FORMAT SPECIFYING MOST OF THE CONVERSION PARAMETERS SETTINGS

Variable	Setting	Description
Sourcekind	Waveform	Source parameterization
Sourceformat	Nist	File format of source
Targetkind	Mfcc_E	Parameterizing Mel-frequency cepstral with log energy
Targetformat	Htk	File format output
Targetrate	100000.0	Target frame rate in 100ns units
Savecompressed	True	Save the output file in compressed form
Windowsize	250000.0	Analysis window size in 100ns unit
Use Hamming	True	Using Hamming window
Preemcoef	0.97	Set a pre-emphasis coefficient
Numchans	26	Number of filtersbank channel (for cestral analysis)
Ceplifter	22	Cepstral liftering coefficient (for cepstral windowing)
Numceps	12	Number of cepstral coefficients
Enormalise	True	Normalize log energy

Acoustic Models

In model training the feature vectors are matched with reference patterns, which are called acoustic models. The reference patterns are usually HMMs trained for whole words, or more often for phones as linguistic units. The goal of training HMMs is to find the most likely model (most likely sequence of states) given the training data, but since the sequence of states itself is hidden, a special iteration method of training is needed to estimate the most likely sequence and then to use the new model to re-estimate the most probable state sequence. HMMs parameters in HTK are estimated using Baum-Welch (B-W) algorithm, an instance of the Expectation-Maximization (EM) algorithm. The idea behind the Baum-Welch algorithm is to recursively calculates the probability of having generated an observation sequence at a particular time (the forward probability) and the

probability of generating the observation sequence from that time until the end (the backward probability). The product of the forward and backward probabilities represents the probability of generating the observation sequence.

HTK supplies four basic tools for parameter estimations: **HCompV, HInit, HRest, and HERest**. HCompV and HInit are used for initialization. HCompV will set the mean and covariance of the entire data set as well as a minimum variance vector which is used to prevent having variance go to zero. Alternatively, a more detailed initialization is possible using HInit which will compute the parameters of a new HMM using Viterbi style of estimation. HRest and HERest are used to refine the parameters of existing HMMs using B-W re-estimation. To fully train a level of model, 2-5 passes of HERest should be made. Once a fully trained monophone model exists, the training data can be realigned using HVite (which is described below) to take multiple pronunciations of words into account [4].

HTK tool **HHEd** is a script driven editor for manipulating sets of HMM definitions. It is mainly used for applying tying across selected HMM parameters; it has facilities for cloning HMMs, clustering states and editing HMM structure. The set of mono-phones models generated previously can then be cloned to create tied-state tri-phone models using this model-editing tool. After performing a few training iterations, the result is a significantly better model. The clustering of model is done to deal with the disk space problem caused by the transformation of mono-phones to tri-phones and to prevent problems with sparse data for tri-phones that occur rarely or never.

Once the models are clustered, the output distributions can be made more complex. HHEd can be used to increase the numbers of Gaussian mixtures per HMM model state. Increasing the Gaussian mixture is achieved by making a copy of Gaussian with the largest weight, then shifting its mean so that it is different from the original and then retraining the model. This has proven to vary the performance of recognition.

Language Models

HTK supports the standard ARPA MIT-LL text format for backed-off N-gram language models. It can be used to develop word-pair grammars, bigram language model, or simple word networks that are specific to a particular recognition task. The HTK tool **HLStats** can generate bigram language models (LMs), **HBuilt** can convert the matrix bigram file from HLStats to an HTK lattice file, **HParse** can generate word level lattices and to test a language model, **HSGen** can generate random utterances given a language model and the number of desired sentences.

4.3.3 Recognition and Analysis

HTK provides a single recognition tool called **HVite** that uses the Viterbi algorithm for pattern matching to perform the recognition and force alignment. HVite takes as input an HMM set, a word label file or a word network (language model), a dictionary, and unknown parameterized speech. If a label file is used instead of a language model, force alignment is performed [4]. The output of HVite is the transcriptions from the matching of a speech file and a network of HMMs.

HResults is the HTK performance analysis tool. It reads in a set of label files (output from a recognition tool such as HVite) and compares them with the corresponding reference transcription files. HResults calculates the percentage correct for sentences and words as well as the percentage accuracy for words.

4.4 Base-line Front-end

Feature extraction is known as front-end of the speech recognition system. There are several front-end methods that are utilized by most speech recognition systems. The chosen front-end for base-line results in this study uses the same parameters kind (MF-PLP) as those used by the HTK group in the DARPA'02 evaluation [5]. These evaluations involved testing of speech recognition systems developed by different leading research group in speech recognition. Among different testing speech data was GSM

speech that was done in [6]. This was part of the motivation behind the choice of these parameters kind. This front-end consist of the following:

- Bandwidth reduced to 125-3800 Hz
- 12 MF-PLP cepstral features + Log Energy term (E) and the 1st and 2nd order derivatives
- Cepstral mean normalization
- Vocal Tract normalization in both training and testing

MF-PLP parameter kind was first used by the HTK group in combination with zeroth cepstral coefficient (C0), 1st and 2nd derivatives, Side-based cepstral mean and variance normalization and vocal tract length normalization as front-end in their 1996 Broadcast News Transcription System and it was found to be more robust and consistent under mismatch conditions than both MFCC (which are currently the method of choice for many Automatic Speech Recognition (ASR) systems) and simple PLP [7]. This showed robustness of MF_PLP parameters from other researchers that motivated even more for its use in the base-line of this study.

In [8, 9] it was found that the loss of information when wide bandwidth clean speech is transmitted through a telephone network is mainly due to the band-limiting nature of telephone network. This showed that the effect of the telephone network could be effectively simulated by simply band-limiting wide-bandwidth clean speech. Band-limiting clean speech was found to increase robustness of clean speech recognition systems to telephone speech. In [8] it was concluded that in some cases band-limited speech recognition systems produce lower word error rates than telephone speech trained recognition systems even when tested with real telephone speech. This motivated the use of band limiting in the base-line which contributed in stopping any unwanted high and low frequencies.

Because Cepstral mean normalization (CMN) is the standard and has shown to be more robust in dealing with channel distortion, it will be the base-line channel estimate in this

study. In our implementation, CMN was computed in both training and testing data to remove their cepstral mean. Vocal tract length normalizations (VTLN) a simple speaker normalization technique derived from the fact that different speakers have different vocal tract lengths. The major concern when implementing VTLN are the choices of warping factors, implementing the warp factor to the data and the warp type (linear, non-linear or piece-wise). In our implementation, the VTLN warp type used is piece-wise defined with a warping factor chosen as the one that gave the best performance per gender in both the training and the testing data. The warping factors chosen for male and female were calculated separately and brought improvements in robustness to real GSM data. The warp factor is applied on the mel-filterbank as suggested in [10, 11, 12].

4.5 Statistical Significance

In experimental design of speech recognition systems where the performance of one design is compared to the performance of another, it is tempting to conclude that if one system has a lower word error rate (WER) on a common task, then it must be better. However, there are many places in an experiment design in which noise can be introduced (e.g., computational noise from parallel processing, fluctuation due to small evaluation data set). Thus, there is a need for statistical measure that can help to determine if an experiment result is statistically significant. The *matched pairs test* and *McNemar's test* are two tests suggested by Gillick and Cox [13] for testing the statistical significance of any differences in performance between two designs. Of the two presented, matched pairs test is the most suitable for continuous speech recognition experiments. This is because the method is best used in experiments whose output can be divided into segments and errors in one segment are supposed to be independent of errors in another segment. In a continuous speech recognition output, these segments are utterances in a test set. The following is the description of the matched pairs test method.

The test involves the difference in numbers of errors of two systems in each segment. The mean of the error differences for all segments is determined. After normalizing by the

estimated standard deviation, this value has an approximately standard normal distribution for a sufficiently large number of total segments (n > 50). See [13].

More specifically, let

$$Z^i = N_A^i - N_B^i \ , \ i = 1, 2 \ldots n \tag{4.1}$$

where N_A^i is the number of errors in the i'th utterance for system A, and N_B^i is the number of errors in the i'th utterance for system B. Then let $\hat{\mu}_z$ be the average difference between errors made by algorithms A and B.

$$\hat{\mu}_z = \sum_{i=1}^{n} Z^i / n \tag{4.2}$$

The estimate of the variance of Z^i is then given by,

$$\hat{\sigma}_z^2 = \frac{1}{n-1} \sum_{i=1}^{n} (Z^i - \hat{\mu}_z)^2 \tag{4.3}$$

Then the test statistic is defined as

$$W = \frac{\hat{\mu}_z}{\left(\hat{\sigma}_z / \sqrt{n} \right)} \tag{4.4}$$

The null hypothesis asserts that the distribution of error differences has mean zero (two-tailed) or has mean no larger than zero (one-tailed, with System B a possible improvement on System A). The null hypothesis is then rejected if the measured value w of W is such that

$$Prob\ (|W| \geq |w|) = 2*Prob\ (W \geq |w|) \leq 0.05\ (two\text{-}tailed) \quad (4.5)$$

$$Prob\ (W \geq w) \leq 0.05\ (one\text{-}tailed) \quad (4.6)$$

The matched pairs test with one-tailed distribution is used in this study to determine the confidence with which any notable improvement brought about can be accepted.

4.6 Baseline Results and Discussions

The performance of the systems in this study are all evaluated in word error rates (WER) defined as,

$$W = \frac{D + I + S}{N} \times 100\% \quad (4.6)$$

Where D is the number of word deletion errors made by the system, I is the number of insertion errors, S is the number of substitution errors and N is the total number of words in the test set used. WER is a better measure of performance than simply comparing the numbers of identical words in the reference and hypothesized transcription since there might be words that are accidentally correct. In the example below, comparing the identity of the words only would give a recognition rate of 100%, whereas the WER is a more representative 50% since there are three substitutions out of a total of six words.

Reference: THE DOG ATE THE CAT'S FOOD
Hypothesis: THE DOG FOOD ATE CAT'S THE

Base-line results, obtained when the base-line system was tested with clean speech and real GSM speech with different number of Gaussian mixture per HMM state (1 - 9), are

shown in Figure 4.5. It can be seen clearly the degradation in performance of the system on GSM speech compared to clean speech from the gap of the curves. The best performance of the system on clean speech occurred at the 7^{th} mixture with the WER of 7.2% and for real GSM speech occurred at the 5^{th} mixture with the WER of 30.53%. Any improvement brought by the proposed method under investigation in this study is evaluated against this WER (30.53%).

4.7 Summary

In this chapter the design of the base-line together with the procedure used in building it have been discussed. This procedure is followed in building all the other systems for experimentation in this study. In the next chapter the Normalization techniques including the theory, experiments and results are discussed.

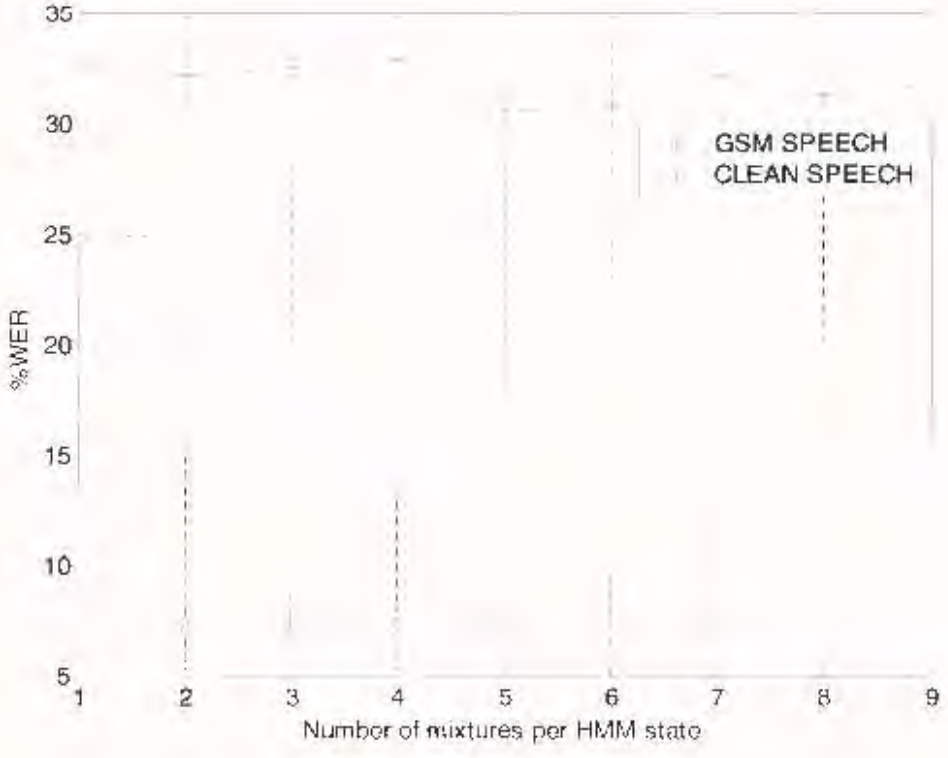


Figure 4.5: Performance of the base-line system on both clean speech and GSM speech.

Chapter 5

Normalization techniques: Theory, Experiments and Results

This chapter discusses some of the most relevant techniques in channel compensation including channel normalization and techniques for reducing variability between speakers. Experiments for testing their effectiveness in compensating for channel effects in real GSM speech are considered. The results of these tests are listed and comparatively discussed.

5.1 Channel Normalization

Channel normalization (CN) techniques have been studied for quite different conditions. In one such condition, which is not discussed in this study, a recognizer is trained with speech recorded with a close microphone and recognition is attempted on speech recorded with a different microphone. In such a case the channel conditions during test are different from those during training, introducing mismatch conditions. These differences must be accounted for explicitly. It is considered good practice to reduce the amount of variation in data as much as possible. To reduce such sources of variations, channel normalization is applied (see for example [26, 27, 28, 29]). In all state-of-the-art automatic speech recognition applications at least over the telecommunication channels, apply some kind of channel normalization to eliminate or compensate for the channel

effects on speech, in order to improve recognition performance of speech affected by a voice communication channel is done.

5.1.1 Channel normalization Background

Voice communication channels tend to exhibit properties of a Linear Time Invariants (LTI) filter acting on speech signal as it propagates through the channel. This filter-like property of the channel distorts speech signals as they are transmitted through the channel. There are noises that are additive to the transmitted speech associated with the channel. This means that the effects of voice communication channel noise on speech signals are either convolutional, additive, or both. Therefore speech transmitted over a communication channel is often expressed as in Equation 5.1 below,

$$y(t) = s(t) * h(t) + n(t) \qquad (5.1)$$

where * is convolution, $h(t)$ is the channel transfer function, $s(t)$ is the transmitted speech signal and $n(t)$ is the channel additive noise.

Most current channel normalization techniques deal with either the channel transfer function distortions or additive noise. Some techniques are designed to deal with both effects of the voice communication channel. There are generally three classes of channel normalization techniques: filter based (pre-processors), feature modification and model modification techniques [4]. Filter based techniques often operate on the signal prior to feature extraction. This is meant to condition a channel speech signal to somehow reduce channel effects imposed on the speech. Feature modification techniques seek to transform extracted features such that the effects of the channel are eliminated. Model modification techniques manipulate parameters of acoustic models to reduce channel effects on speech recognition.

Some of the well-known channel normalization techniques are reviewed in brief. These techniques address the following problems:

- Channel distortion
- Additive noise
- Both channel distortion and additive noise

Cepstral mean normalization (CMN) is perhaps one of the most effective algorithms because of its simplicity [30, 31, 32, 36]. The technique is commonly used in speech and speaker recognition for normalization of different microphones, as well as telephone channels. It was originally proposed by Atal [59]. Its Fourier spectrum is expressed by the Equation 5.2 below:

$$Y(f) = S(f)H(f) + N(f) \qquad (5.2)$$

If the logarithm is taken on Equation 5.2, the term $S(f)H(f)$ becomes,

$$\ln S(f)H(f) = \ln S(f) + \ln H(f) \qquad (5.3)$$

The Equation 5.3, shows that the convolution between original signal spectrum $S(f)$ and channel transfer $H(f)$ becomes simple addition in the logarithm spectral domain which is known as cepstral domain. This means in the cepstral domain, the channel transfer function is an additive component. Therefore it can be subtracted from the cepstral features.

In CMN, the algorithm takes advantage of the relationship in Equation 5.3. This states that the contribution of the channel is additive, and computes along-term mean value of the feature vectors. Then is subtracts this mean value from each of the vectors in the cepstral domain. This way it ensures that the mean value of the incoming feature stream is zero.

In doing so, the variability of the data is reduced and allows for a simple and yet effective channel and speaker normalization. The procedure is applied to both the training and testing data. For a better estimate of the cepstrum of the transfer function as a mean, a lot of data is required to compute. CMN works well for unknown channels, where a system is trained with speech obtained by one channel while testing is done using speech obtained from a different channel. The only weakness is that it doesn't deal with additive noise; it only addresses convolutional noise as expressed by the equation-based analysis and produce a distortion in the presence of additive noise [4].

In two independent studies, the effectiveness of two well known channel normalization techniques, RelAtive SpecTrAl fitering (RASTA) and CMN, were compared in a recognition set-up based on context independent HMMs [32, 37]. In these two studies, the task was the recognition of digit strings. Different languages were being used: [32] used German and American English, while [37] used Dutch. In both studies CMN was found to be more effective. The only weakness of CMN is that it doesn't deal with additive noise, but only addresses convolutional noise.

CMN is the normalization choice in this study for dealing with the channel effects. This method is chosen due to its simplicity, popularity and effectiveness in speech recognition as shown by research.

RelAtive SpecTrAl fitering (RASTA) was developed by Hermansky and Morgan [34]. Since speech is produced by movements of vocal tract at different changing rates, these movements and their rates of changes are reflected in speech components (linguistic components) of recorded speech signals. Non-speech components of such signals often have rates of change that lie outside the range of linguistic components [34]. RASTA processing takes advantage of these differences. It aims to suppress spectral components of a recorded speech signal that change more slowly or quickly than the normal rate of linguistic components [34]. Because the filtering is in the cepstral domain, RASTA deals with convolutional noise and not additive noise. RASTA processing was developed as an extension of the PLP feature extraction method.

RASTA and CMN are techniques for dealing with channel distortion and both are inexpensive. However, CMN has shown to be more effective. Results in [40] found that RASTA needs detailed models to work well, i.e. the models cannot be context-independent. Hanson and Applebaum [38, 39] compared RASTA and CMN and concluded that CMN performs better than RASTA.

Further developments in RASTA techniques have allowed for additive noise compensation at low SNR. Adaptive Lin-Log RASTA [34, 40] is the modification of regular RASTA processing, and is referred to as J-RASTA

Mean and Variance Normalization (MVN) is a further development in CMN technique. It is the combination of CMN and a normalization of the variance (hence MVN). It improves the compensation of the mismatch with respect to CMN, and an improvement of the recognition performance. In a study of improving the insensitivity of the feature vector, it is well known, that a constant, though unknown channel transfer function, affects the mean of the cepstral features. Further it has been observed that additive noise results, among other effects, in a mean shift and reduction of the variance of the distributions of the cepstral coefficients [60]. Hence it is a good ideal to perform both mean and variance normalization. The effect of variance normalization is that, irrespective of the dynamic range of the input feature stream, each output feature has unit variance (and power, because of cepstral mean normalization).

In further studies [6, 61], CMN and MVN provided significant improvements for cepstral based representations and MVN showed to be slightly better than CMN. However MVN method is limited because it cannot compensate for the non-linear effects caused by the additive noise.

Maximum likelihood channel estimate as presented by Neumeyer, Digalakis and Weintraub [41] as well as Sankar and Lee [42, 43] can be subtracted from a set of cepstral observation vector. This method is accurate but computationally expensive

compared to CMN. It is a method with a channel estimate that maximizes the probability of the utterance, assuming a fixed but unknown stationary channel for each utterance. Different studies for effectiveness of ML channel estimation and CMN were compared. In two independent studies, the percentage word error for CMN was found similar to that of the ML channel estimate but slightly higher [44, 45].

Codeword-Dependent Cepstral Normalization (CDCN) [47] models the distributions of cepstra of clean speech by a mixture of Gaussian distributions. It analytically models the effect of the noise and channel vectors on the distributions of clean speech cepstrum. The algorithm works in two steps. The goal of the first step is to estimate the values of the noise and channel vectors that maximize the likelihood of the observed noisy cepstrum vector. In the second step, cleaning the data is applied to find the unobserved cepstral vector of clean speech given the cepstral of noisy speech. The algorithm works on a sentence-by-sentence basis, needing only the sentence to be recognized to estimate noise and channel vectors. These steps increased the already significant computation for using CDCN. CDCN procedure is attractive because it can simultaneously compensate for the effect of additive noise and linear filtering. CDCN and RASTA were investigated in [46] and CDCN showed to be slightly more effective in the case of training with clean speech and testing using speech over the telephone network, although its performance over the telephone network is worse than in other noisy environments.

5.1.2 Channel normalization by CMN

In order to develop telephone or mobile application using speech recognition, basic work in dealing with the channel distortion and noise is needed. CMN from previous research has proven to be the simplest and most effective channel normalization of them all and it is currently the method of choice in speech recognition. This method was the choice in this study. It was applied on two tested feature extraction methods, the MFCC and MF-PLP. The two front-ends are similar to the base-line front-end outlined in Section 4.4 except that they have neither channel normalization (as CMN) nor Speaker Normalization (as VTLN), which is yet to be discussed.

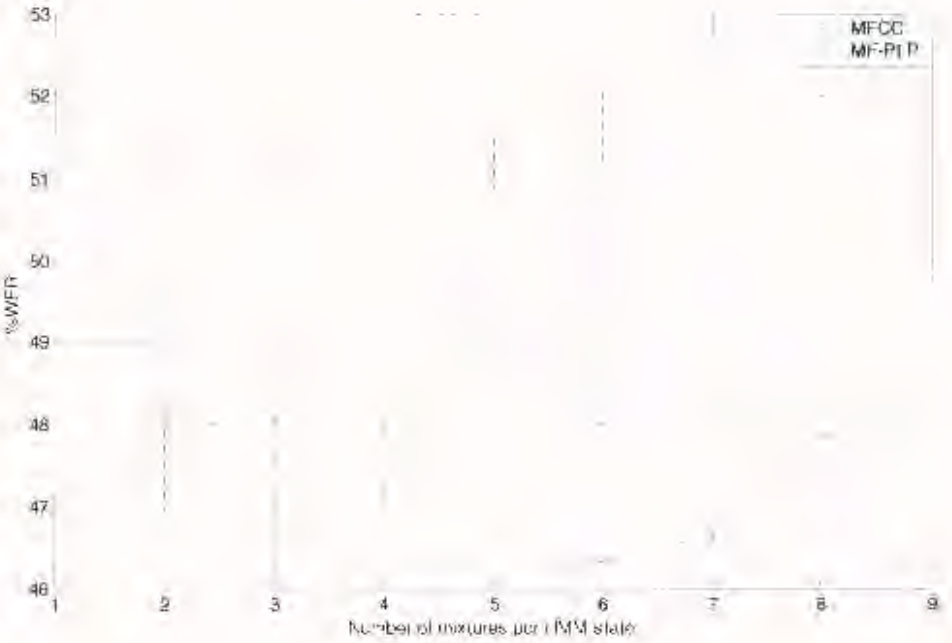


Figure 5.1: Results on recognition of system with MFCC and MF-PLP feature extraction methods tested without any normalization technique

Experimental Results and Discussions

In this section we present the results for recognition system built without any normalization technique and then the results on the system with CMN technique applied for channel compensation. The CMN results for the experiments using MFCC and MF-PLP feature extraction methods are presented in Figure 5.2 and for system built without any normalization are presented in Figure 5.1.

No Normalization technique

Figure 5.1 presents the results obtained from two feature extraction methods tested without any normalization.

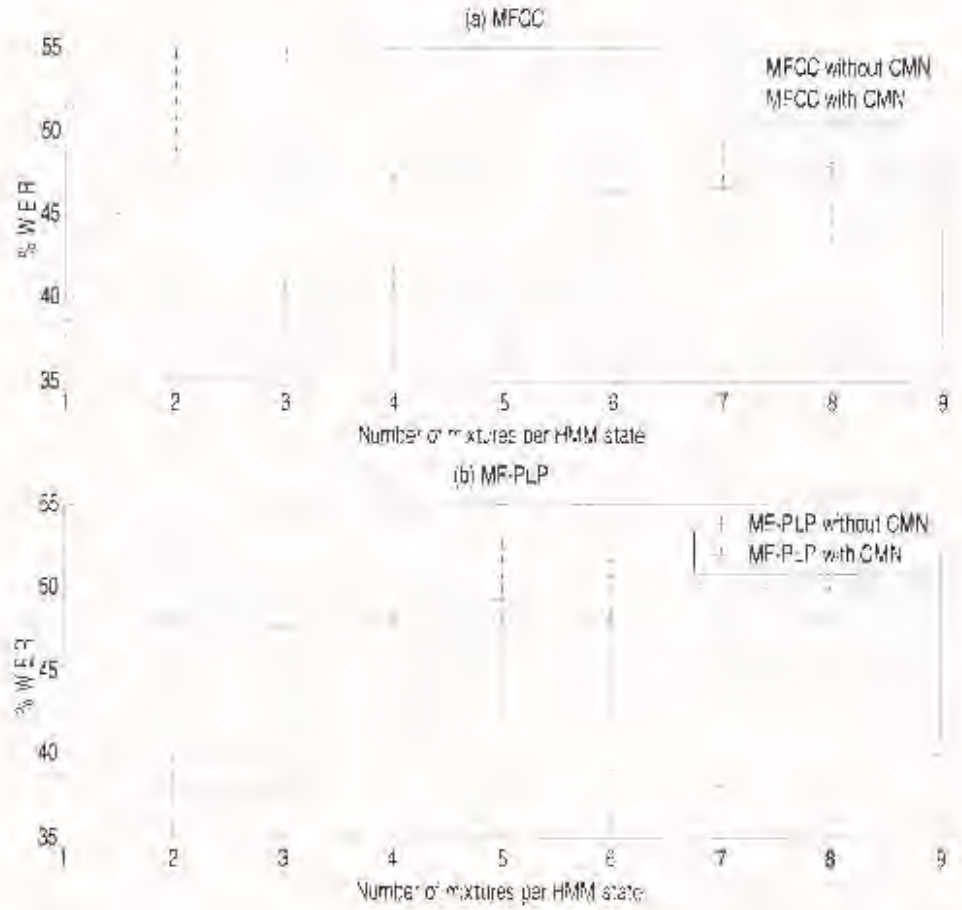


Figure 5.2: Results on recognition with CMN applied on (a) MFCC and (b) MF-PLP feature extraction methods.

The lowest WER produced by the MFCC is 46.14% at the 5th mixture and for MF-PLP it is 47.61% at the 3rd mixture. These results show that MFCC is more robust to GSM speech than MF-PLP in our basic system. Comparison was also performed on simulated GSM speech on three different feature extraction methods [62, 63], the PLP, MFCC, and MF-PLP. The results showed that the PLP was more robust to GSM speech than the MFCC and MF-PLP. However these results were not consistent and could not be relied upon. In these results, MF-PLP performed slightly better than MFCC which is not the case in this study. In their system, in place of, the log energy E which was the parameter coefficient appended to the target kind in this study, the zero'th cepstral parameter coefficient was appended which might have contributed to the difference in performance.

CMN

Figure 5.2 presents the results obtained when CMN is applied on the two feature extraction methods tested. It can be seen clearly from the results that CMN reduces the WER for both methods. The lowest WER produced by MFCC is 36.27% at the 6^{th} mixture, which was a 21.4% drop from the lowest result produced by the system on a same method without CMN. On the other hand MF-PLP produced its lowest WER at the 8^{th} mixture with a 36.33%, which is 23.7% drop from the lowest result produced by the system on the same method without CMN.

From the results presented above and other research findings, the dominance and effectiveness of CMN is evident [32, 37]. CMN reduced the WER significantly in both feature extraction methods and resulted in a small difference between the two systems.

5.2 Speaker Normalization

Almost all speech recognizers are to some extent sensitive to the variation of the speaker and speaker's environment. The performance of speech recognition system could vary largely in practical use because of these variations. Therefore, to make speech recognition systems as accurately as possible and at the same time as robust as possible is a major issue in speech recognition. In this study a method of speaker normalization accomplished by a transformation of the frequency axis based on speaker specific acoustic features is presented. This speaker normalization method attempts to increase speech recognition accuracy by reducing those speakers's variability. Speaker normalization for the past years has been used in conjunction with speaker adaptation for a speaker dependent systems since the two strategies tackle the variations from speaker, channel, and environment [54, 55, 56].

5.2.1 Speaker Normalization Background

Speaker dependent systems, which come from the speaker dependent speech signal, are known to out perform speaker independent systems when enough training data are available. Speaker variability accounts for much of this difference in performances. Thus, reducing this variability can decrease this difference in performance. Speaker variability appears for different reasons; mostly to do with external influences related to linguistic differences like speakers cultural background, emotional state, etc., and physiological differences between speakers such as difference in shapes and size of components of the vocal tract [48, 49]. But it is generally agreed that one major source of inter-speaker speaker variance is the vocal tract shape, especially the vocal tract length (VTL) [50, 51]. Therefore, some researchers have been devoted to the vocal tract length normalization (VTLN) for speaker normalization [48, 49, 50, 51, 52, 53].

Differences in vocal tract dimensions will cause differences in spectra of the sound, even when the speaker is producing a sound that we perceive as the same phoneme. The warping of the frequency axis is an attempt to modify spectra so that the distance between the spectra of sounds perceived as the same phoneme is smaller. Previous attempts in speaker normalization [49, 52, 53] have mostly relied upon "trial and error" approach: the algorithm tests several "solutions" for the warping function and select one producing the best results.

Speaker normalization can be performed using linear and nonlinear frequency warping function. The selection of the use between linear and nonlinear warping functions is based on maximization of likelihood, and sometimes based directly on speaker specific parameters. Warping function in the context of speaker normalization is a function mapping two spectra. If for instance, one intend to map the spectrum $X(w)$ to spectrum $Y(w)$ one can use a function $f(w)$ so that,

$$\hat{Y}(w) = X(f(w)) \tag{5.2}$$

The effect of the $f(w)$ will be an expansion or compression of the spectrum $X(w)$ depending on whether the first derivative of f, $f'(w)$, is bigger or smaller than the unit.

Recently, Andreou *et al.* [57] proposed a set of maximum likelihood based speaker normalization procedures to extract and use acoustic features that are robust to variations in vocal tract length. The procedure reduces speaker dependent variations between formant frequencies through a simple linear warping of frequency axis. The warping function for a given speaker is iteratively chosen as one that maximizes the likelihood of hypothesis transcription at the output of the decoder. While this and other studies of frequency warping procedures have shown improved speaker independent automatic speech recognition (ASR) performance, the performance improvements were achieved at the cost of highly computationally intensive procedures [58]. A number of similar studies followed that of Andreou *et al.*, and most of them proposed faster alternatives to selection based on decoder outputs. For example, Wegmann *et al.* [53] proposed an algorithm where the warping function for a particular speaker is chosen from a set of warping function, based on maximization of likelihoods associated with a Gaussian mixture model that statistically represents the standard speaker. The warping factor ranging of the warping function proposed by Wegmann *et al* is between 0.88 and 1.12.Several other techniques have been proposed to make frequency warping methods even more efficient with less computationally procedure.

5.2.2 Speaker Normalization by VTLN

The VTLN is a well-known approach to speaker normalization, which aims at reducing speaker specific variations of the speech signal caused by different lengths of the vocal tract. The motivation behind VTLN is the fact that positions of spectral formant peaks for utterances of a given sound are inversely proportional to the vocal-tract length. There are three major factors considered when VTLN is implemented and these are, the choice of a warping factor, how to apply the warping factor to data and the warp type (i.e. linear, non-linear). In the implementation of VTLN in this study the warp type used is piece-

wise linear defined with a warping factor α chosen as the one that gave the best performance for speakers.

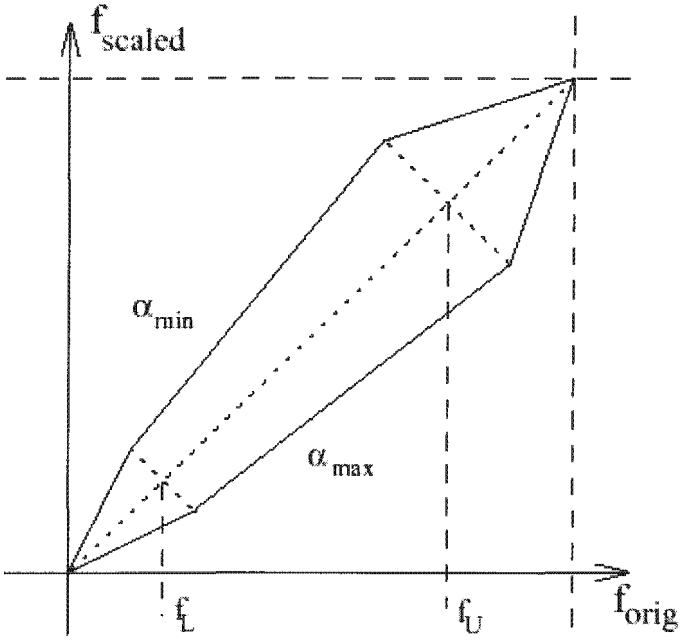


Figure 5.3: Frequency Warping [3]

Here values of $\alpha < 1.0$ corresponds to compressing the spectrum, and $\alpha > 1.0$ correspond to stretching the spectrum, and $\alpha = 1.0$ corresponds to no warping case. Figure 5.3 shows the overall shape of the resulting piece-wise linear warping functions.

In the Figure 5.3, the range of the warping factor is reflected. The lower and upper curves correspond to the maximum and minimum factors respectively, and the middle dotted line corresponds to unit warping factor (no warping). This range of warping factor for piece-wise is very limited, from 0.8 to 1.2. The f_{orig} axis is of the unwarped frequency and f_{scale} axis is of the warped frequency. As the warping would lead to some filters being placed outside the analysis frequency range, the simple linear warping

function is modified at upper and lower boundaries. The result is that the lower boundary frequency f_L and the upper boundary frequency f_U are always mapped to themselves.

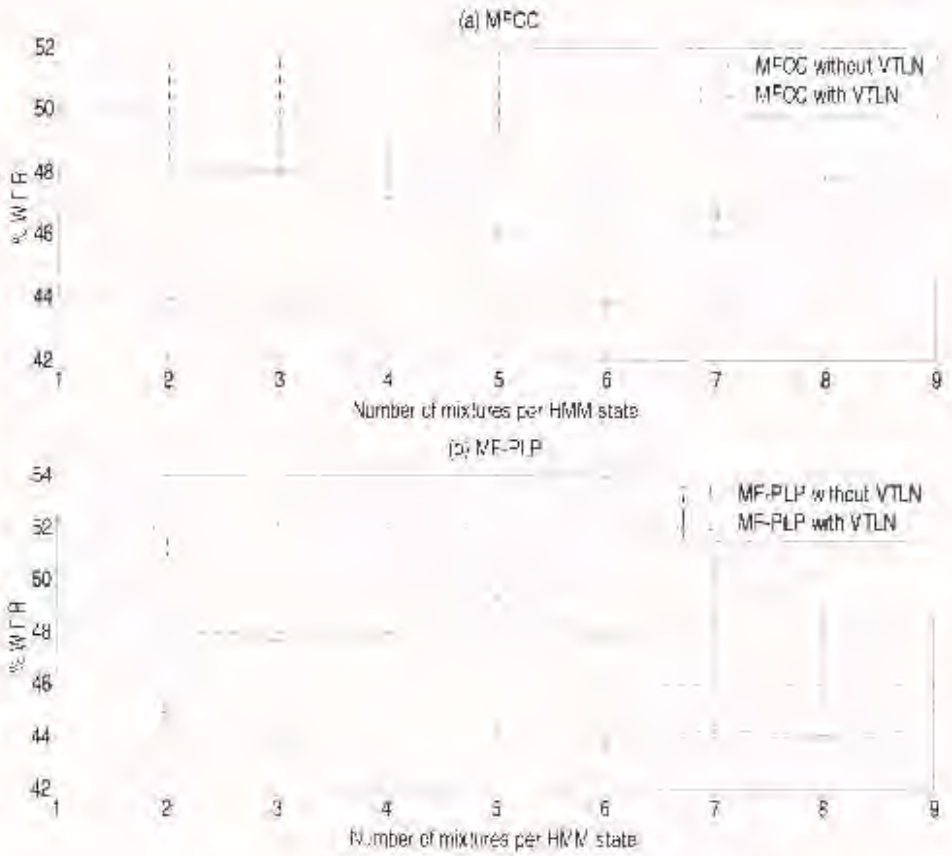


Figure 5.4: Results on recognition with VTLN applied on (a) MFCC and (b) MF-PLP feature extraction methods

The need for estimation of warping factor for each speaker imposes computational cost. This issue of computational cost led to the idea of using only two warping function one for each gender, which greatly reduced the cost. Since females usually have tracts with smaller dimension than male, therefore values that gave the best performance for each gender were used. The warp factor is applied on the mel-filterbank as suggested in [52, 53, 54]

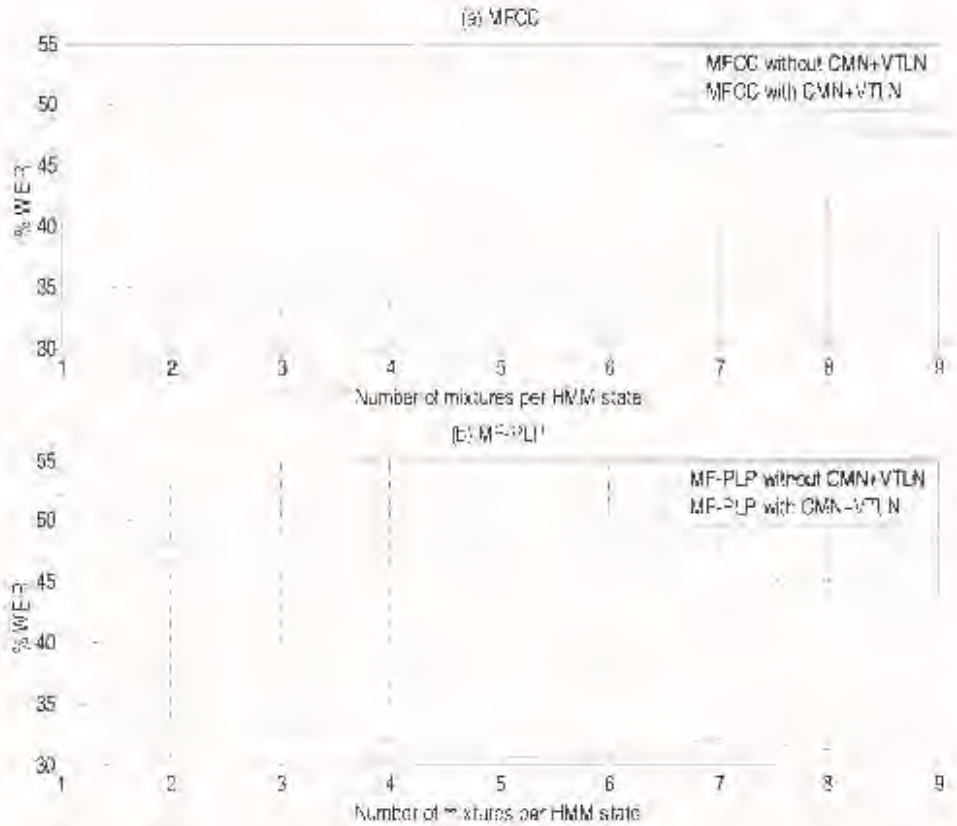


Figure 5.5: Results on recognition with the combination of CMN and VTLN applied on (a) MFCC and (b) MF-PLP feature extraction methods.

The lowest WER produced by MFCC is 32.7% at the 8th mixture, which was a 29.1% drop from the lowest result produced by the system on a same method without CMN and VTLN. On the other hand MF-PLP produced its lowest WER at the 5th mixture with a 30.53%, which is 35.9% drop from the lowest result produced by the system on the same method without CMN and VTLN.

In recent studies [6, 66, 67] similar combination have been investigated were MVN technique, which is a further development of CMN as mentioned in Section 5.1.1, was used with VTLN to evaluate the robustness of the front end with MFCC, MF-PLP and PLP feature extraction methods. In their experiments, they showed that the combination

was effective and resulted in better performance with MF-PLP, which is a similar finding in this study.

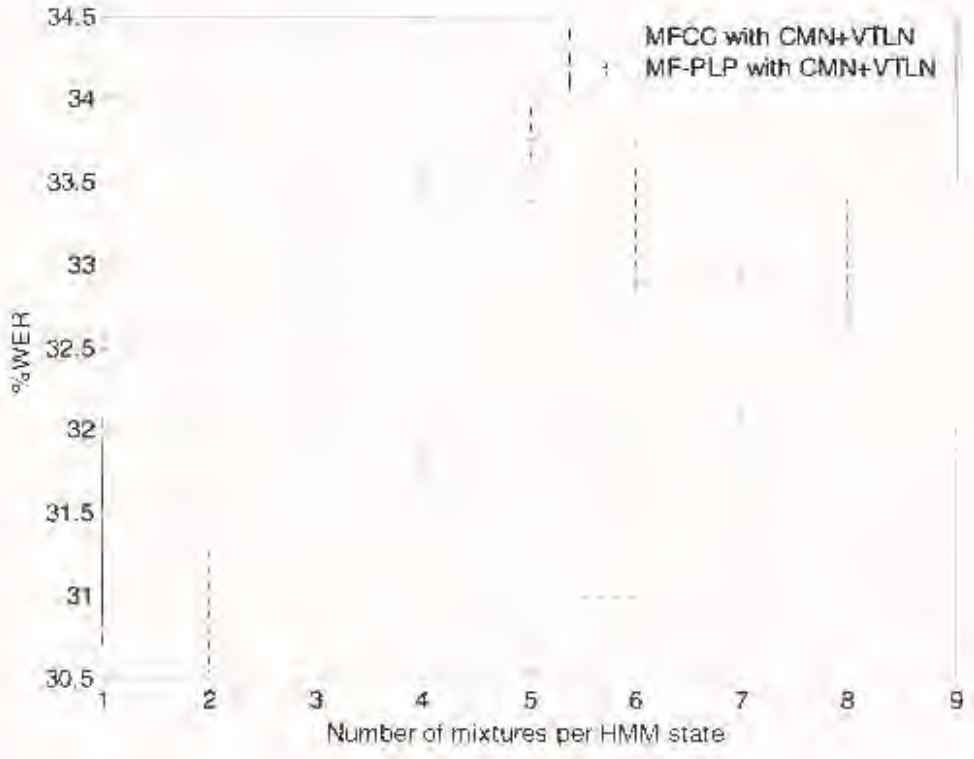


Figure 5.6: A comparison of performance on MFCC and MF-PLP feature extraction methods on combination of CMN and VTLN.

Form the results it is clear that channel normalization technique CMN reduces the recognition word error rate and proved to be effective and speaker normalization technique VTLN has managed to account for the effect of the variation in vocal tract length and reduced recognition word error rate. In Figure 5.6 the comparison in performance of the two baseline systems is shown. The baseline system with the MF-PLP feature extraction method in combination with CMN and VTLN produced better overall results than the baseline system that used MFCC features in combination with CMN and VTLN. Since MF-PLP produced better results, it was chosen to be the baseline front-end to be implemented for further improvement.

5.4 Summary

In this chapter, concepts of channel normalization and speaker normalization are briefly discussed. The theory behind CMN and VTLN were discussed in detail. Experiments were conducted with CMN and VTLN separately and combined to determine if they brought about any significant improvements on real GSM speech recognition performance and to compare their performance. From the results obtained it is clear that channel normalization technique CMN reduced the recognition word error rate and showed to be effective and Speaker normalization technique VTLN has managed to account for the effect of the variation in vocal tract length and reduced the recognition word error rate. The combination of these methods further improved recognition accuracy for GSM speech recognition system in all tested front-ends, the MF-PLP and the MFCC. MF-PLP produced better results under this combination; hence it was chosen to be the base-line front-end. In the next section the pre-filter implemented in combination with CMN and VTLN intended to further improve the real GSM speech recognition performance of the base-line system is discussed and results of the experiments are reported.

Chapter 6

Noise Suppression Filter: Theory, Experiments and Results

This chapter begins with motivation for the experiments conducted with the noise suppression filter followed by the discussion of the theory behind the development of this noise suppressor method. Lastly, experiments conducted with this filter together with the results are mentioned and discussed.

The noise suppression filter proposed in this study to improve robustness of recognition systems to GSM speech is widely used in enhancing noisy speech. It is also useful in the arena of low-bit rate digital communication. Its success in telecommunication is clearly seen in the fact that different variations of it have been incorporated into several speech coding standards as a speech enhancement pre-processor [76].

Among these standard speech coders is the Federal standard 2.4 kbps MELP vocoder. Since the first US and North Atlantic Treaty Organization (NATO) standard LPC-10 speech coder, low bit-rate speech coding has greatly improved at the 2.4 khps data rate. New vocoders provide increased intelligibility and quality but nevertheless are sensitive to background noise and hence the role of noise suppression in speech coding gains importance in an effort to increase the speech SNR at the input to the codec. The use of speech enhancement techniques as a pre-processing stage to speech coders is being

applied to governmental and military wireless communication systems. This approach was with a goal of improving the perceptual quality of degraded speech that is encoded with 2.4 kbps voice coder [76]. The filter in this study is used for the same purpose, to improve perceptual quality of coded speech by reducing perceived noise introduced by low-bit speech coder and the background noise on GSM speech.

6.1 Motivation for Experiments

In the GSM standard no post filtering is used for improving the quality of the speech. This is because the perceptual quality of the speech is considered good enough even without employing a speech enhancement post-filter after a speech has been transferred through the network. However, good perceptual quality does not mean good quality speech for speech recognition.

Knowing the possibility of a mobile phone owner being able to use the same phone almost anywhere on the planet, the number of locations in which a handset can be used is large, and each location will have its own particular noise environment. In such cases, the quality of speech the listener receives can be poor, and difficult to understand. This is even worse for speech recognition systems. If the background noise can be reduced to an extend that it doesn't interfere with the speech, the ability for better speech quality and better speech recognition would be greatly enhanced.

Knowing that neither the channel normalization method Cepstral Mean Normalization (CMN) nor the speaker normalization method Vocal Tract Length Normalization (VTLN) used in this study compensate for the additive noise. The use of a speech enhancement method to compensate for the additive noise would be of a great benefit. It is for this reason that in this study, the noise suppression filter Minimum Mean-Square Error Log- Spectral Amplitude Estimator (MMSE-LSA) which is used as a speech enhancement method is used as a pre-filter to speech recognition systems. This is an attempt to improve the quality of GSM speech and in the process improving the

recognition thereof. Its function is to reduce GSM speech coding and background noise prior to recognition.

6.2 Noise suppression techniques

In the past, the topic of noise suppression has been used in an attempt to enhance signals of various kinds including speech. In [68], some of the earliest known work on adaptive noise cancellation (ANC) was presented. In general the application of adaptive noise cancellation algorithms requires the use of two-channels. A primary channel which receives the corrupted speech input and a secondary channel for the reference noise source. In a non-stationary environment, single-channel systems can be used for filtering [69]. As the name implies, single-channel enhancement techniques are only applicable in cases where there is one acquisition channel, (e.g. a telephone channel), where as multiple-channel enhancement techniques are applicable only when there are several acquisition channels, (e.g. a microphone array configuration). Since this research study seeks to improve the recognition of GSM speech. It is therefore limited to single-channel enhancements concepts. The main criterion for the noise suppression filtering is choosing an algorithm to determine the manner in which the filter coefficients are updated. The most common of these algorithms is the Minimum or Least Mean Squares Error (MMSE, LMSE) estimation, which minimizes the mean power of the output [69].

A comprehensive overview and summary of various noise suppression techniques can be found in [70]. The listed methods include Spectral Subtraction, Wiener estimation, Maximum-likelihood estimation, soft-decision method which in most cases operates in the frequency-domain. The basic problem that these filter address is estimation of the magnitude of speech from the noisy observation.

Inherent between the noise suppression algorithms is the computational complexity. For instance, in the spectral subtraction algorithm, the Short-Time Spectral Amplitude (STSA) is estimated as the square root of the Maximum-Likelihood (ML) estimator of each signal spectral component variance. In systems exploiting Wiener filtering, the

STSA estimator is obtained as the modulus of the optimal MMSE amplitude estimator of each signal spectral component. A common feature of these techniques is that the noise reduction process brings about very unnatural artifacts called "musical noise".

Both Spectral Subtraction and Wiener Filtering are well known and commonly used speech enhancement techniques. These have been applied to a wide variety of enhancement situations. Some of the situations include enhancing and improving performance of the low bit-rate speech codecs in a form of preprocessing to increase the speech signal to noise ratio. Such an approach is simple in that it does not require any modification of the speech-coding algorithm. Recently, significant improvements have been reported when specific speech coders were combined with speech preprocessor. Guilmin et al [77] showed that Wiener filter-based noise preprocessing significantly improved the output, in the presence of noise, of a low rate LPC vocoder both in terms of parameter estimates and subjective quality. Earlier Kang and Fransen [78] evaluated spectral subtraction enhancement for LPC-processing of noisy speech and reported dramatic improvement in subjective quality in speech corrupted with variety of background noise.

However, there are severe underlying limitations to the amount of noise that can be removed by Spectral Subtraction and Wiener Filtering. The aim of most enhancing algorithms is to remove most of the noise leaving comfortable residual noise with minimal speech distortion. Removing more noise than is necessary can produce distorted outputs. However, removing less noise can have the counter effect of suppressing weak energy phonemes. These distortions and residual noises are critical when the input SNR is low. The most common introduced residual noise is Musical Noise.

In 1984, Ephraim and Malah [71] derived an MMSE-STSA estimator that assisted in the reduction of the annoying musical noise. A study on the elimination of musical noise is presented in [79]. The gain computed for each in their algorithm is based on the probability of speech absence and is computed for each frequency bin. Later in 1985, they published a paper, [72], where the MMSE of the Log-Spectral Amplitude (MMSE-

LSA) is used in enhancing noisy speech. In either case, the enhanced speech sounds very similar but this new estimator was found to be more effective in enhancing the noisy speech, and significantly improves its quality [72]. Recently a number of different preprocessing schemes were examined for use with the Federal standard 2.4 kbps MELP coder using MMSE-LSA [76].

Most of these noise suppression schemes have been designed to increase the quality, insofar as intelligibility and naturalness of corrupted speech are concerned. It is worthwhile to see research done in joint systems (speech enhancement and low-bit rate coder similar to those used in GSM coding) in improving the quality of the speech prior to recognition. The Ephraim and Malah MMSE-STSA and MMSE-LSA noise suppression rule have been reported to attenuate the musical phenomenon considerably without introducing audible distortion. It also proves to perform better than Spectral subtraction and Weiner filter [71]. Hence the MMSE-LSA speech enhancement noise suppression filter is proposed as a pre-filter for removing noise in real GSM speech with a view of improving the perceptual quality of the speech.

6.3 MMSE Log-Spectral Amplitude Estimator

A year before Ephraim and Malah proposed the MMSE-LSA algorithm, they had suggested an algorithm for enhancing speech degraded by uncorrelated additive noise when the noisy speech alone is available. This algorithm capitalized on the major importance of the short-time spectral amplitude (STSA) of the speech signal in its perception, and utilized a minimum mean-square error (MMSE) STSA estimator for enhancing the noisy speech [71]. The goal of the STSA algorithm is to estimate the modulus of each complex Fourier expansion coefficient of the speech signal in a given analysis frame of noisy speech. Research has shown that a distortion measure that is based on the mean-square error of the log-spectral is more suitable for speech processing see [73]. This widespread log-spectra in distortion measure is the leading motivation to examine the effect of STSA estimator. This estimator minimizes the mean-squared error

of the log-spectra in enhancing noisy speech. The derivation of the MMSE LSA is summarized below.

6.3.1 Derivation of the MMSE-LSA Estimator

Let *x(t)* and *d(t)* denote the speech and the noise processes, respectively. Then the observed noisy signal, *y(t)* is given as:

$$y(t) = x(t) + d(t) \qquad 0 \le t \le T-1 \tag{6.1}$$

The objective of a speech enhancement is to estimate *x(t)* which is accomplished on a frame-by-frame basis by applying a unique gain to each of the frames of *y(t)*. These gains are computed in either frequency or time domain by minimizing or maximizing Signal-to Noise Ratio (SNR).

Let $X_k \overset{\Delta}{=} A_k \exp(j\alpha_k)$, $Y_k \overset{\Delta}{=} R_k \exp(j\vartheta_k)$ and D_k denote the *k*th spectral component of the original clean signal *x(t)*, the noise *d(t)* and the noisy observation *y(t)*, respectively, in the analysis frame interval [0, T-1]. A_k, R_k and D_k denote the spectral magnitude of clean signal, noisy signal and the noise only observation while α_k and ϑ_k denote the phase of clean and the noisy signal respectively. Y_k (and similarly X_k and D_k) are given by:

$$Y_k = \sum_{t=0}^{T-1} y(t) \exp\left(-j\frac{2\pi}{N}kt\right) \qquad k = 0, 1, 2 \ldots T\text{-}1 \tag{6.2}$$

Based on these definitions, the problem reduces to estimating the modulus of X_k from the degraded signal $\{y(t),\ 0 \le t \le T-1\}$.

According to the definitions given above in Equation 6.1 and their respective Fourier transforms, the driving constrain on finding the optimal realization, $\hat{A}_k$, of the logarithmic A_k, is expressed as:

$$\hat{A}_k = E\{(\log A_k - \log \hat{A}_k)^2 \mid Y_k\} \tag{6.3}$$

Hence, the estimator is easily shown to be:

$$\hat{A}_k = \exp\{E[\ln A_k] Y_k\} \qquad 0 \le t \le T-1 \tag{6.4}$$

and it is independent of the basis chosen for the log. The evaluation of Equation 6.4 can be simplified with the *moment generating function,* [74]. For notational convenience, let $Z_k \overset{\Delta}{=} \ln A_k$. Then the moment generating function $\Phi_{z_k|Y_k}(\mu_k)$ of Z_k given Y_k can be expressed as:

$$\Phi_{z_k|Y_k}(\mu_k) = E\{\exp(j\mu_k Z_k) \mid Y_k\} = E\{A_k^{j\mu} \mid Y_k\} \tag{6.5}$$

The first derivative of $\Phi_{z_k|Y_k}(\mu_k)$, evaluated at $\mu = 0$ generates the first moment of Z_k given Y_k [74] as seen in Equation 6.6 below.

$$E[\ln A_k] Y_k\} = \frac{d}{d\mu} \Phi_{z_k|Y_k}(\mu_k) \mid_{\mu=0} \tag{6.6}$$

Therefore, our task is to calculate $\Phi_{z_k|Y_k}(\mu_k)$ and then to obtain $E[\ln A_k] Y_k\}$ by using Equation 6.6. From Equation 6.5, $\Phi_{z_k|Y_k}(\mu_k)$ is given by:

$$\Phi_{z_k|Y_k}(\mu_k) = E\{A_k^{j\mu} \mid Y_k\}$$

$$= \frac{\int_0^{\infty}\int_0^{2\pi} a_k^{u} p(Y_k \mid a_k, \alpha_k) p(a_k, \alpha_k) d\alpha_k da_k}{\int_0^{\infty}\int_0^{2\pi} p(Y_k \mid a_k, \alpha_k) p(a_k, \alpha_k) d\alpha_k da_k} \tag{6.7}$$

Since the observed data is assumed to be Gaussian distributed, Equation 6.7 can be expanded in terms of $p(Y_k \mid a_k, \alpha_k)$ and $p(a_k, \alpha_k)$:

$$p(Y_k \mid a_k, \alpha_k) = \frac{1}{\pi \lambda_d(k)} \exp\left\{ -\frac{1}{\lambda_d(k)} \mid Y_k - a_k e^{j\alpha_k} \mid^2 \right\} \tag{6.8}$$

$$p(a_k, \alpha_k) = \frac{a_k}{\pi \lambda_x(k)} \exp\left\{ -\frac{a_k^2}{\lambda_x(k)} \right\} \tag{6.9}$$

where $\lambda_x(x) \stackrel{\Delta}{=} E\{|X_k|^2\}$, and $\lambda_d(k) \stackrel{\Delta}{=} E\{|D_k|^2\}$, are defined as variances of the kth spectral component of the speech and noise, respectively. Simple substitution of Equation 6.8 and Equation 6.9 in Equation 6.7, and using the integral representation of the modified Bessel function of zero order $I_0(.)$ [75, eq. 8.406.3, 8.411.1], the following equation is obtained.

$$\Phi_{z_k|Y_k}(\mu_k) = \frac{\int_0^{\infty} a_k^{\mu+1} \exp(-a_k^2/\lambda_k) I_0(2a_k\sqrt{v_k/\lambda_k}) da_k}{\int_0^{\infty} a_k \exp(-a_k^2/\lambda_k) I_0(2a_k\sqrt{v_k/\lambda_k}) da_k} \tag{6.10}$$

where λ_k satisfies the following relation

$$\frac{1}{\lambda_k} = \frac{1}{\lambda_x(k)} + \frac{1}{\lambda_d(k)} \tag{6.11}$$

and v_k is defined as:

$$v_k = \frac{\varepsilon_k}{1+\varepsilon_k}\gamma_k \tag{6.12}$$

The parameters, ε_k and γ_k from Equation 6.12 are interpreted as *a priori* SNR and *a posteriori* SNR values, respectively:

$$\varepsilon_k \stackrel{\Delta}{=} \frac{\lambda_x(k)}{\lambda_d(k)}; \quad \gamma_k \stackrel{\Delta}{=} \frac{R_k^2}{\lambda_d(k)} \tag{6.13}$$

A priori SNR is the Signal-to-Noise Ration of the *k*th spectral component of the "clean" speech signal, *x(t),* while *a posteriori* SNR is the kth spectral component of the corrupted signal, *y(t).* Computation of γ_k, a *posteriori* SNR, is a straightforward ratio of the variance of noisy speech signal to the estimate noise variance. However, computation of ε_k, *a priori* SNR, is more involved especially that the knowledge of "clean" signal is rarely available in real system. The two approaches taken to compute a *priori* SNR are "*Decision-Directed"* estimation and Maximum Likelihood estimation [71]. The integrals in Equation 6.10 are then evaluated by using [75, eq. 6.631.1, 8.406.3, 9.212.1] and yield the following equation.

$$\Phi_{z_k|Y_k}(\mu_k) = \left(\lambda_k^{\mu/2}\Gamma(\mu/2+1)\right)\left(\sum_{r=0}^{\infty}\frac{(-\mu/2)_r}{(1)_r}\frac{(-v_k)^r}{r!}\right) \tag{6.14}$$

where $\Gamma(.)$ is the Gamma function. Hence the derivative of Equation 6.14 that is needed in Equation 6.6 is given by the following equation as shown in [72, 75].

$$\frac{d}{d\mu}\Phi_{z_k|Y_k}(\mu_k)\Big|_{\mu=0} = \frac{1}{2}\ln\lambda_k + \frac{1}{2}\left(\ln v_k + \int_{v_k}^{\infty}\frac{e^{-t}}{t}dt\right) \tag{6.15}$$

simple substitution of Equation 6.15 into Equation 6.6 and using Equation 6.12 and 6.4 gives the desired amplitude estimator in Equation 6.16. This equation defines the gain function for MMSE-LSA in Equation 6.17.

$$\hat{A}_k = \frac{\varepsilon_k}{1+\varepsilon_k}\left\{\frac{1}{2}\int_{v_k}^{\infty}\frac{e^{-t}}{t}dt\right\} R_k \tag{6.16}$$

Then,

$$G_{LSA}(\varepsilon_k, \gamma_k) = \frac{\hat{A}_k}{R_k}$$

$$= \frac{\varepsilon_k}{1+\varepsilon_k}\left\{\frac{1}{2}\int_{v_k}^{\infty}\frac{e^{-t}}{t}dt\right\} \tag{6.16}$$

Ephraim and Malah did some comparisons between MMSE-STSA and MMSE-LSA to show that the enhanced speech (using the latter estimator) suffers from much less residual noise, while there is no perceptible difference in the enhanced quality of speech itself [72]. In order to explain the phenomenon, one should examine the parametric gain curves of the presented estimator MMSE-LSA (G_{MMSE}) and compare them to those produced by MMSE-STSE estimator (G_{LSA}) as shown in Figure 6.1. According to the figure, MMSE-LSA offers more attenuation (or lower gain values) for the corresponding instantaneous SNR than its counterpart [72]. This is easily seen by the analyzing Jensen's inequality as shown in Equation 6.17.

$$\hat{A}_k = \exp\{E[\ln A_k \mid Y_k]\} \le \exp\{E[A_k \mid Y_k]\} = E[A_k \mid Y_k] \tag{6.17}$$

This Equation 6.17 indicates that the log estimate of $\hat{A}_k$ is always less than or equal to the true estimate.

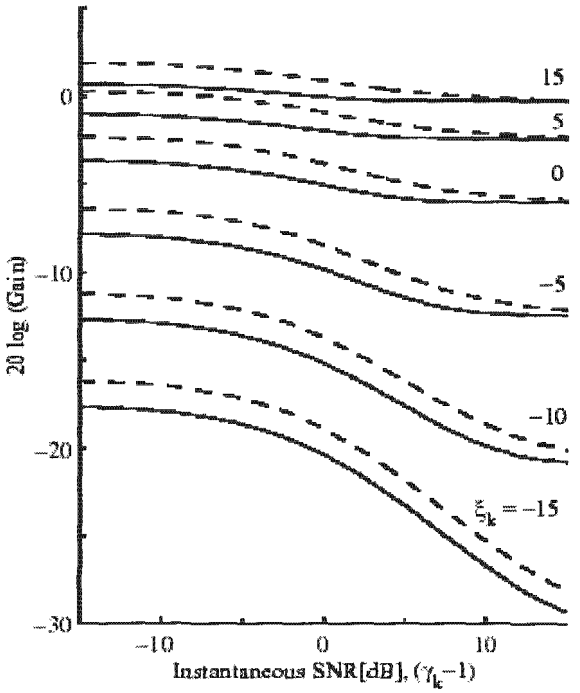


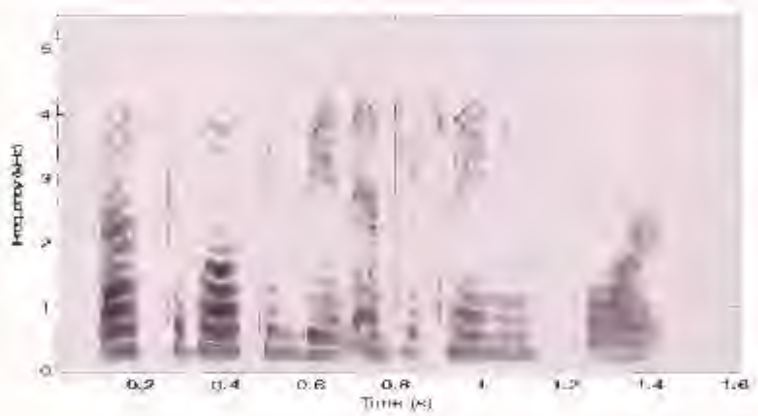
Figure 6.1: Parametric gain curves describing G_{MMSE} defined by [71, eq. (7)] (solid lines) and G_{LSA} defined by Eq. 6.16 (dashed lines) for various values of ε_k (in dB) [72].

Since, MMSE-LSA works in the log domain to estimate the magnitude of A_k, as oppose to MMSE-STSA algorithm, it confirms why the MMSE-LSA gain plots are lower than those of the MMSE-STSA in Figure 6.1. These lower values of gain curves further minimize the effects of residual noise especially at low instantaneous signal to noise ratio values.

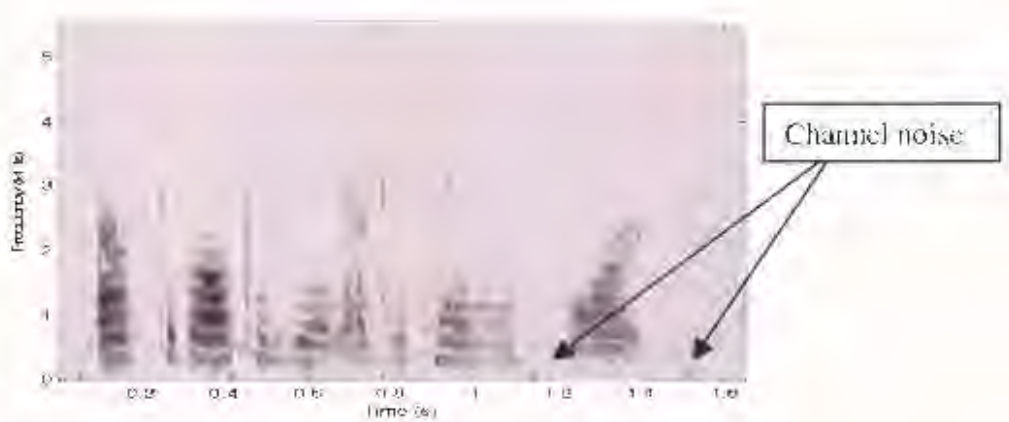
6.4 Experimental Results and Discussion

In this section, the results of experiments conducted with the noise suppression filter are presented and discussed. The experiments were conducted and compared with the base-

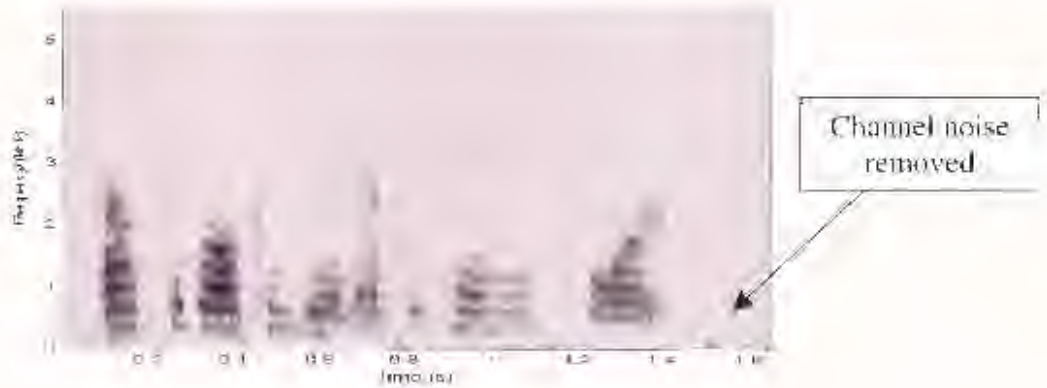
line. Performance results of applying the speech enhancement MMSE-LSA noise suppression filter as a pre-filter are first displayed using spectrograms. This is a way of comparing performance of pre-filter as the amount of channel noise reduction that occurs. Figure 6.2 presents different spectrograms of speech "*Basketball can be an entertaining sport*". Figure 6.2(a) presents the original clean speech, Figure 6.2(b) presents the original GSM speech and Figure 6.2(c) presents the GSM speech after enhancement. In Figure 6.2(c), shows after application of an enhancement process, the speech nearly resembles the original clean speech given in Figure 6.2(a). Comparing Figure 6.2(b) and Figure 6.2(c), it is clear that there is a reduction of the corrupting channel noise hence showing that the filter has improved the perceptual quality of GSM speech.



(a)



(b)



(c)

Figure 6.2: Spectrograms of (a) Original clean Speech (b) Original GSM Speech (c) GSM Speech after enhancement

Having seen the influence of the filter on the GSM speech from the spectrogram, recognition performance results are then presented in Figure 6.3. These results show that the filter brings about further improvements with the word error rate (WER) lower than that produced by the base-line system. This improvement is visible in most of the number of mixtures experimented in this study.

The best %WER produced by this system is 28.49 % at the 4th mixture. This is a 6.7 % reduction of the WER produced by the base-line. This is a total of 40.2% drop in WER comparing to 47.61% WER produced by case with no channel compensation to 28.49% WER produced by the proposed system. Using the matched pair test this gave a confidence of 90 % in accepting that the filter has improved the performance of the tested system given real GSM test data.

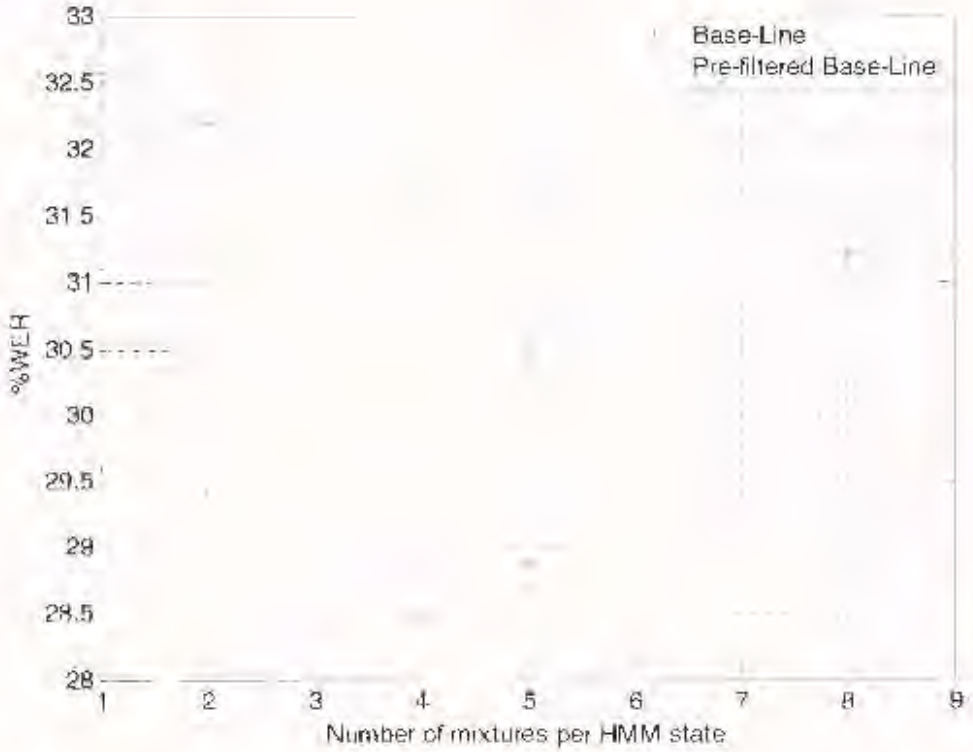


Figure 6.3. A comparison of performance of the system with the pre-filter and the baseline

6.5 Further Experiments on Proposed system

The MF-PLP recognition system was used as base-line system for this study because of its better performance. However, the filter was also investigated as the pre filter using MFCC recognition system on GSM speech. This is because MFCC feature extraction method is well known and considered as the state of the art feature extraction method prior to speech recognition. Results of this experiment are shown on Figure 6.4. The performance of the filter was evaluated against the values presented on Figure 5.4(a) of the MFCC speech recognition with the combination of CMN and VTLN.

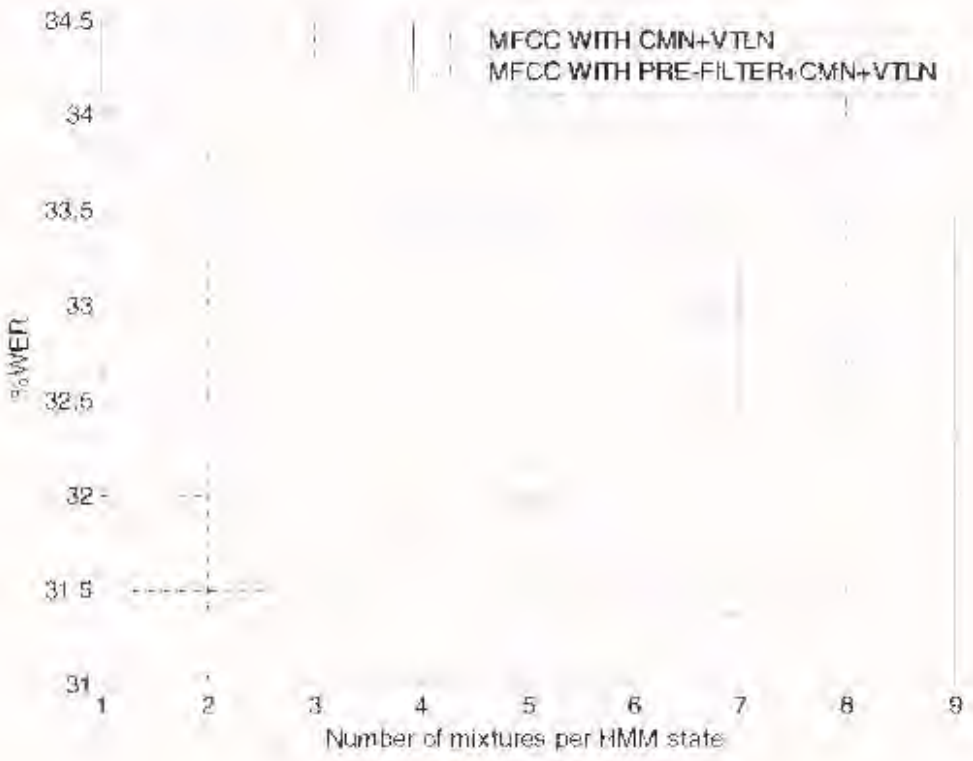


Figure 6.4: A comparison of performance of the system with the pre-filter and the system with out on MFCC

The improvement is visible in most of the numbers of mixtures experimented. The best %WER produced by this system using the pre-filter is 31.33 % at the 3rd mixture compared to 32.70% produced by the system without a pre-filter. The improvement brought about by the use of the filter on MFCC recognition system as a pre-filter on GSM speech is about 4.2% reduction in WER. This is a total of 32.1% drop in WER produced when comparing 46.14% WER produced by case with no channel compensation to 31.33% WER produced by the proposed system. Using the matched pair test this gave a confidence of 71.2 % in accepting that the filter has improved the performance of the tested system given real GSM test data. This shows that the filter can improve perceptual quality and hence improve recognition in different types of feature sets.

More studies of the proposed system for recognition of real GSM speech were conducted. These include investigating if the system can be generalized, and if it is stable in terms of its performance. Different environments with similar characteristics as GSM were used. Telephone network (land-line) speech, which is a telecommunication channel as GSM speech, was used. NTIMIT database that was created by transmitting sentences in the TIMIT database over telephone line was used for this evaluation.

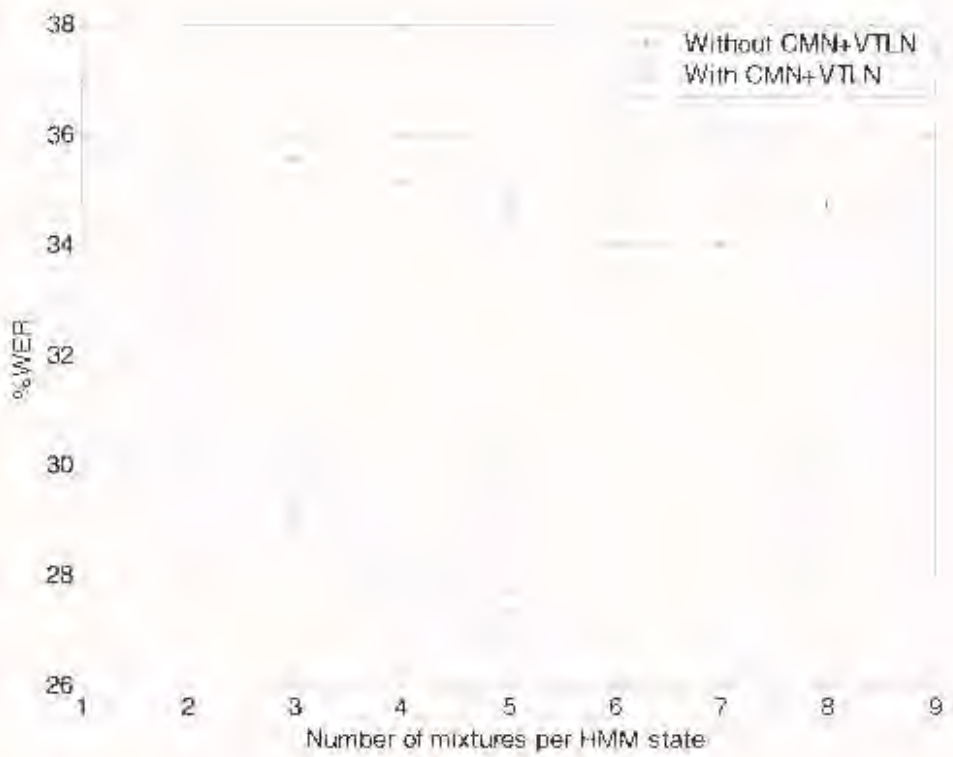


Figure 6.5: Results on recognition with CMN and VTLN and system with the combination of CMN and VTLN applied on MF-PLP feature extraction methods for telephone speech.

Using the same procedure used in establishing a base-line for GSM speech, the results are presented on Figure 6.5. The combination of CMN and VTLN to form the base-line system can be seen from results that it reduces the WER significantly on the proposed feature extraction methods (MF-PLP). These results are compared to the results of a

system built without any normalization technique. The lowest WER produced is 26.83% at the 6th mixture which was a 21.2% drop from the lowest result produced by the system with the same method without CMN and VTLN. The improvement brought by pre-filtering is evaluated against this WER.

Lastly an experiment was conducted to test the pre-filter in combination with CMN and VTLN for telephone speech. This system is similar to the one, proposed in this study for GSM speech recognition. The results of this experiment are shown on Figure 6.6. The lowest WER produced by the system using pre-filter is 25.05% at the 6th mixture compared to 26.83% produced by the system without pre-filter. This is a 6.6% reduction in WER brought by the pre-filter. This is a total of 26.4% drop in WER produced when comparing 34.03% WER produced by case with no channel compensation to 25.05% WER produced by the proposed system. Using the matched pair test this gave an 89.25% confidence that the filter has improved performance of the tested system, given Telephone data.

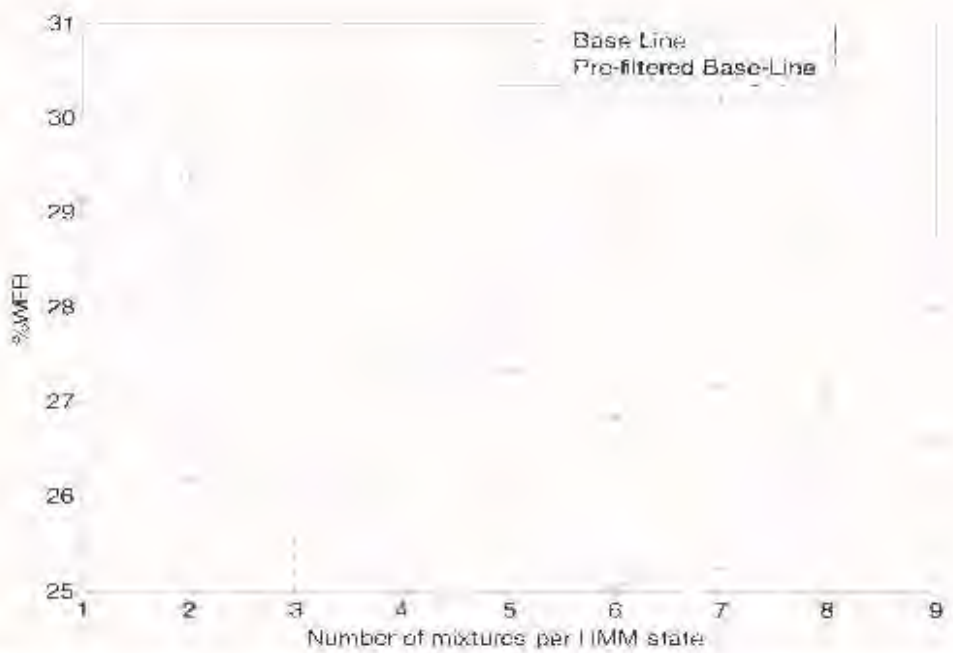


Figure 6.6: A comparison of performance of the system on Telephone speech with the pre-filter and the baseline on MFPLP

From these further experiments, it is evident that the proposed system does improve recognition performance. Therefore, it can be used for recognition of different kinds of feature sets and on telephone speech as well, hence the proposed system can be a generalized.

6.6 Summary

In this chapter, the noise suppression techniques have been discussed. The definition and derivation of the proposed pre-filter was discussed. The results of the experiments conducted with the pre-filter are then mentioned and discussed. It was found that the filter enhances speech recognition robustness to GSM speech by reducing the channel effects on GSM speech signal. It was also shown that the filter can be used for recognition of different kinds of feature sets and on telephone speech. In the next chapter conclusion drawn from these results and suggested further work is given.

Chapter 7

Conclusion and Future Work

7.1 Summary and Conclusion

Improving the robustness of speech recognition systems to GSM speech was the ultimate goal of this study. The recognition performance of GSM speech is poor due to GSM channel noise and GSM speech coding. Different channel normalization approaches have been developed to address channel effects and improve robustness of speech recognition systems to voice communication channel speech. In this study, the CMN technique was tested to determine its effectiveness with GSM speech. The fact that different speakers have different vocal tract length motivated the development of a speaker normalization technique in an attempt to increase speech recognition accuracy by reducing speaker's variability. VTLN is a simple speaker normalization technique implemented in this study in combination with CMN to determine its effectiveness with GSM speech.

A speech enhancement noise suppression filter widely used in enhancing noisy speech and useful in the arena of low-bit rate digital communication was proposed as a pre-filter. This was with an aim of removing noise in real GSM speech. It improves the perceptual quality of the GSM speech by reducing effects of GSM speech coding on GSM speech recognition performance. The expectation was that this filter combined with channel normalization and speaker normalization techniques would bring higher GSM speech recognition rates if used. The pre-filter was tested in combination with the CMN and VTLN.

The study concluded that the use of noise suppression filter as pre-filter together with CMN and VTLN does improve robustness of recognition systems to GSM speech. The improvement brought about is 40.2% reduction in WER compare to the case where no channel compensation is used with a confidence of 90%.

Second, the study found that an improvement in perceptual quality of speech for which this filter investigated is mainly used does mean improved recognition of the enhanced speech. In addition the filter used as a pre-filter has been used in enhancing noisy speech and in the arena of low-bit rate digital communication to enhance speech perceptual quality. In this study the filter has been used in speech recognition application and it has brought about improvement in robustness to GSM speech.

These conclusions were drawn from the result of the experiment conducted with the filter used as a pre-filter for an MF-PLP recognition system. Even though MF-PLP recognition system was considered the base-line system for this study based on better performance, the filter was investigated using MFCC recognition system since MFCC feature extraction method is well known and considered as the state of the art feature extraction method prior to speech recognition. The improvement brought about by the use of the filter together with CMN and VTLN on MFCC recognition system as a pre-filter is about 32.1% reduction in WER at 71.2% level of confidence. This shows that the filter can improve perceptual quality and hence improve recognition in different type of feature sets.

The filter was tested together with CMN and VTLN under different environment, including telephone speech. It improved performance of telephone speech by about 26.4% reduction in WER at 89.25% level of confidence. This once again shows that if perceptual quality of speech is improved then that would guarantee improvement in recognition performance of the enhance speech.

7.2 Suggestions for Future Work

Based on the study performed the following suggestions for further work are given:

- The filter in this study was tested with two different feature extraction methods (MF-PLP and MFCC). Since different feature extraction methods compute different features of speech, the pre-filter may affect them. Hence it would be interesting to find out how the pre-filter performs with different feature extraction methods.
- Since more studies have been done on channel normalization, including advancing from Cepstral Mean Normalization to Cepstral Mean and Variance Normalization, it would be interesting to apply this advanced technique using the proposed configuration.
- The study should be furthered into the effect of different environments, including background noises, using the proposed configuration.

References:

[1] DAPRA, "the DAPRA TIMIT acoustic-phonetic continuous speech corpus, training and testing data," 1990.

[2] K. Lee, *Automatic Speech Recognition: The development of the SPHINX system.* Boston: Kluwer Academic Publisher, 1989.

[3] S. Young and P. Woodland. (2003) The Hidden Markov Model Toolkit (HTK) version 3.2.1 [Online]. Available: http://www.htk.eng.cam.ac.uk/

[4] R.A. Bates, "Reducing the effects of linear channel distortion on continuous speech recognition," Msc. Dissertation, Harvard Divinity school, Harvard University, 1993.

[5] ____, "CU-HTK April 2002 switch board system," in *Rich Transcription Workshop 2002*, May 2001.

[6] N.S. Mahlanyane, D.J. Mashao, "Using a low-bit Speech Enhancement Adaptive Filter to Improve GSM Speech Recognition Performance," Cape Town. *Proc. SATNAC 2003.* November 2003 pp. 75-79.

[7] P.W. et al.,"The development of the 1996 HTK broadcast news transcription system," in Proc. DAPRA Speech Recognition Workshop, 1997.

[8] M. Weintraub and L. Neumeyer, "Constructing telephone acoustic models from a high-quality speech corpus," in *Proceeding of ICASSP'94*, 1994.

[9] M. Weintraub, V. Digalakis, and L. Neumeyer, "Training issues and channel equalization techniques for the construction of telephone acoustic models using a high-quality speech corpus," IEEE *Transactions on Speech and Audio Processing,* pp. 590-597, October 1994.

[10] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, 49-60, Jan 1998.

[11] D. Pye and P. Woodland, "Experiments in speaker normalization and adaptation for large vocabulary speech recognition," in *Proc of ICCASP'97*, 1997, pp. 1047-1057.

[12] P. Zhan and A. Waibel, "Vocal tract length normalization for large vocabulary continuous speech recognition," in *CMU-CS-97-148*, Carnegie Mellon University, Pittsburgh, PA, May 1997.

[13] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithm," in *Proc. ICASSP'89*, Glasgow, England, May 1989, pp. 532-535.

[14] P. Woodland and S. Young, "The HTK tied-state continuous speech recognizer," in *Proc. Eurospeech'93,* Berlin, July 1993, pp. 2207-2210.

[15] A.S. Spanias, "Speech Coding: A Tutorial Review," *in Proc. IEEE*, Vol. 82, No. 10, October 1994. pp 1541-1582.

[16] R. Salami, L. Hanzo, R. Steel, K. Wong, and I. Wassel, *Mobile Radio Communication*. Pentech Press, London, 1992, ch. Speech Coding.

[17] B. Lilly and K. Paliwal, "Effects of speech coders on speech recognition performance," in *Proc. Int. Conf. Spoken Language Processing (ICSLP'96)*, USA-Philadelphia, 1996, pp. 2344-2347.

[18] F. Itakura and S. Saito, "Analysis-by synthesis theory based on the maximum likelihood method," in *Proc of 6th Int. Congress on Acoustic*, Tokyo, 1968, pp. c17-20

[19] 3G Americas: What is GSM (no date). [On-line]. Available: http://www.3gamericas.org/English/Technology_Center/QA/gsmqa.cfm [2004, May 25]

[20] C. Online. (2004, May) Latest global, handset, base station and regional cellular statistics. [Online]. Available: http://www.celillar.co.za

[21] GSM World: "GSM Technology" (no date). [On-line]. Available: http://www.gsmworld.com/technology/gsm.shtml [2004, May 25]

[22] J. Degener. "GSM 06.10 lossy speech compression" (July 200). [On-line]. Available: http://kbs.cs.tu-berlin.de/~jutta/toast.html [2004, May 25]

[23] L. Besacier, S. Grassi, A. Dufaux, M. Ansorge, F. Pellandini, "GSM Speech coding and speaker recognition," *Proc. of ICASSP'00*, Istanbul, Turkey, June 200.

[24] I. Gerson and M. Jasiuk, "A 5600 bps VSELP speech coder candidate for half rate GSM," *Proc. Eurospeech'93*, Vol. 1, pp. 253-256, 1993.

[25] J. M. Huerta, "Speech recognition in mobile environments," Ph.D. dissertation, Department of Elec. And Comp. Engineering, CMU, 2000.

[26] A. Anastasakos, F. Kubala, J. Makhoul, R. Schwartz, "Adaptation to new microphones using tied-mixture normalization," *in Proc. ARPA Spoken Language Techn. Workshop*, pp. 89-93.

[27] F. Liu, P. Moreno, R. Stern, A. Acero, "Signal processing for robust speech recognition," *in Proc. ARPA Spoken Language Techn. Workshop*, pp. 110-115, 1994.

[28] J. Orloff, L. Gillick, R. Roth, F. Scattone, J. Baker, "Adaptation of acoustic models in large vocabulary speaker independent continuous speech recognition," *in Proc. ARPA Spoken Language Techn. Workshop*, pp. 119-122, 1994.

[29] M. Weintraub, L. Neumeyer, V. Digalakis, "SRI November 1993 CSR spoke evaluation," *in Proc. ARPA Spoken Language Techn. Workshop*, pp. 135-144, 1994.

[30] B. Alta, "Automatic recognition of speakers from their voices," *in Proc. IEEE*, vol. 64, 460-475, 1974.

[31] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust. Speech Signal Process*, vol. 29, 254-272, 1981.

[32] R. Haeb-Umbach, P. Beyerlein, D. Geller, "Speech recognition algorithms for voice control interfaces," Philips J. Res., vol. 49, 381-397, 1995.

[33] H. Hermansky, N. Morgan, A. Bayya, P. Kohn, "Compensation for the effect of the communication channel in auditory-like analysis of speech," *in Proc. Eurospeech*, pp. 1367-1370, 1991,.

[34] H. Hermansky, N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, 578-589, 1994.

[35] J. Koehler, H. Hermansky, N. Morgan, H. Hirsch, G. Tong, "Integrating RASTA-PLP into speech recognition," *in Proc. Internat. Conf. Acoust. Signal Speech Process.*, pp. 421-424, 1994.

[36] V. Steinbiss, H. Ney, X. Aubert, S. Besling, C. Dugast, U. Essen, D. Geller, R. Haeb-Umbach, R. Kneser, H.-G, Meier, M. Oerder, B.-H. Tran, " The Philips Research system for continuous-speech recognition," *Philips J. Res.*, vol. 49, 317-352, 1995.

[37] J. de Veth, L. Boves, "Channel normalization techniques for automatic speech recognition over the telephone," *Speech Communication*, vol. 25, 149-164, 1998a.

[38] B. Hanson, T. Applebaum, "Subband or Cepstral Domain Filtering for Recognition of Lombard and Channel-distorted Speech," *Proc. Internat. Conf. Acoust., Signal Speech Process.*, pp. 1179-1182, 1993.

[39] B. Hanson, T. Applebaum, "Robust Speaker-independent Word Recognition Using Static, Dynamic and Acceleration Feature: Experiments with Lombard and Noisy Speech," *Proc. Internat. Conf. Acoust., Signal Speech Process.*, pp. 857-860, 1990.

[40] H. Hermansky, N. Morgan, "Toward Handling the Acoustic Environment in Spoken Language Processing," *Internat. Conf. Spoken Language Process.* pp. 85-88, 1992.

[41] L. Neumeyer, V. Digalakis, M. Weintraub, "Training Issues and Channel Equalization Techniques for the Construction of telephone Acoustic models Using a High-Quality Speech Corpus," *IEEE Trans. Speech Process.*, pp. 590-597, 1994.

[42] A. Sankar, C. –H. Lee, "A maximum-likelihood approach to stochastic matching fro robust speech recognition," *IEEE Trans. Speech Process.*, 1996.

[43] A. Sankar, C. –H. Lee, "Robust Speech Recognition based on Stochastic matching," *in Proc. Internat. Conf. Acoust. Signal Speech Process.*, pp. 121-124, 1995.

[44] R.A. Bates, M. Ostendorf, "Maximum likelihood channel estimation for WSJ Telephone speech recognition," Boston University Electrical, Computer and Systems Engineering, Technical Report No. ECS-96-002, 1996.

[45] M. Rahim, B. –H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Process.*, vol. 4, no.1, pp. 19-30, 1996.

[46] P.J. Moreno, "Speech recognition in telephone environments," Msc Thesis, Department of Elec. And Comp. Engineering, CMU,

[47] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," Ph.D. Thesis, Department of Elec. And Comp. Engineering, CMU, 1990.

[48] C. Tuerk, T. Robinson, "A new frequency shift function for reducing inter-speaker variance," in *Proc. Eurospeech '93,* vol. 1, pp. 351-354, 1993.

[49] E. B. Gouvea, "Acoustic feature based frequency warping for speaker normalization," Ph.D. Thesis, Department of Elec. And Comp. Engineering, CMU, 1998.

[50] H. Wakita, "Normalization of vowels by vocal-tract length and its application to vowel identification," *IEEE Trans. ASSP,* vol. 25, pp. 183-192, 1977

[51] Y. Ono, H. Wakita, Y. Zhao, "Speaker normalization using constrained spectral shifts in auditory filter domain," in *Proc. Eurospeech '93,* vol. 1, pp. 355-358, 1993.

[52] L. Lee, R. Rose, "Speaker normalization using efficient frequency warping procedures," in *Proc of ICCASP '96*, pp. 353-356, 1996.

[53] S. Wegmann, D. McAllaster, J. Orloff, B. Peskin, "Speaker normalization on conversational telephone speech," in *Proc of ICCASP '96*, pp. 339-341, 1996.

[54] D. Pye, P. Woodland, "Experiments in speaker normalization and adaptation for large vocabulary speech recognition," in *Proc of ICCASP '97*, pp. 1047-1050, 1997.

[55] J. McDonough, T. Schaaf, A. Waibal, "Speaker adaptation with all-pass transforms," *Speech Communication*, vol. 42, 75-91, 2004.

[56] R. Hariharan, O. Viikki, "An integrated study of speaker normalization and HMM adaptation for noise robust speaker-independent speech recognition," *Speech Communication*, vol. 37, 349-361, 2002.

[57] A. Andreou, T. Kamm, J. Cohen, "Experiments in vocal tract normalization," *in Proc. CAIP Workshop*: *Frontiers in Speech Recognition II*, 1994.

[58] L. R. Rabiner, B. –H. Juang, "Fundamentals of speech Recognition," Englewood Cliffs, NJ: Prentice-Hall, 1993.

[59] B. Alta, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," Journal of the Acoustical Society of America, vol. 55, 1304-1312, 1974.

[60] J.P. Openshaw, J.S. Mason, "On the limitations of Cepstral Features in Noise," in *Proc of ICASSP'94*, pp. II49-II52, Adelaide, Australia, 1994.

[61] P. Jain, H. Hermansky, "Improved mean and variance normalization for robust speech recognition," in *Proc. of ICASSP 2001,* Salt Lake City, 2001.

[62] P. Woodland, M. Gales, D. Pye, "Improving environmental robustness in large vocabulary speech recognition," in *Proc. of ICCASP'96,* Atlanta, GA, 1996, pp. 65-68

[63] P. Woodland, M. Gales, "The HTK large vocabulary recognition system for the 1995 arpa h3 task," Available: citeseer.nj.nec.com/132495.html

[64] E.B Gouvea, "Acoustic-Feature-Based frequency warping for speaker normalization," Ph.D. Thesis, Department of Elec. And Comp. Engineering, CMU, 1998.

[65] P. Zhan, A. Waibel, "Vocal tract length normalization for large vocabulary continuous speech recognition," CMU-CS-97-148, Carnegie Mellon University, Pittsburgh, PA, 1997.

[66] P. Woodland, M. Gales, D. Pye, S. Young "The development of the 1996 HTK broadcast news transcription system," this is a report on research work conducted by the CU-HTK group.

[67] P.W.et al., "The development of the 1996 HTK broadcast news transcription system," in *Proc. DAPRA Speech Recognition Workshop,* 1997

[68] B. Windrow, J. R. G. Jr, J. M. McCool, J. Kaunitz, R. H. H. C S Williams, J. R. Zeidler, E. D. Jr, R. C. Goodlin, "Adaptive noise canceling: Principle and applications," in *Proc. of IEEE*, vol. 63, pp. 1692-1717, 1975

[69] M. Tuffy, "The Removal of Environmental Noise in Cellular Communications by Perceptual Techniques," Ph.D. Thesis, Department of Electronics and Electrical Engineering, University of Edinburgh, 1999

[70] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," in *Proc. of IEEE*, vol. 67, pp. 1586-1604, 1979

[71] Y. Ephraim, D. Malah, "Speech enhancement using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol ASSP-32, pp. 1109-1121, 1984

[72] Y. Ephraim, D. Malah, "Speech enhancement using a Minimum Mean Square Error Log-Spectral Amplitude Estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol ASSP-33, pp. 443-445, 1985

[73] R. M. Gray, A. Buzo, A. H. Gray, Jr., Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol ASSP-28, pp. 367-376, 1980

[74] A. L. Garcia, *Probality and Random Processes for Electrical Engineering*. Reading: Addison-Wesley Publishing Company, second ed., 1993

[75] I. S. Gradshteyn and Z. M. Ryzhik, *Table of Integrals, Series, and Products*. New York: Academic, 1980

[76] T. Agarwal, "Pre-processing of noisy speech for voice coders," Msc Thesis, Department of Electrical and Computer Engineering, McGill University, Montreal, Canada, 2002.

[77] G. Guilmin, R. Le Bouquin-Keanns, P. Gournay, "Study of the influence of noise pre-processing on the performance of a low bit rate parametric speech coder," in *Proc. Europ Conf. on Speech Comm. And Tech*, vol. 5, pp. 2367-2370, 1999.

[78] G.S Kang, L.J. Fransen, "Quality improvement of lpc-processed noisy speech by using spectral subtraction," *IEEE Trans. On Acoustic, Speech and Signal Processing*, vol. 37, no. 6, pp. 939-942, 1989.

[79] O. Cappe, "Eliminiation of the Musical Noise Phenomenon using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Trans. On Acoustic, Speech and Signal Processing*, vol. 2, pp. 345-349, 1994

[80] L. Baum, T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," in *Ann. Math. Stat.*, vol. 37, pp. 1554-1563, 1966

[81] L. Baum, J. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360-363, 1967

[82] L. Baum, G. Sell, "Growth functions for transformations on manifolds," *Pacific Journal of Mathematics*, vol. 27, no.2, pp. 211-227, 1968

[83] L. Baum, G. Soules, N. Weiss, T. Petrie, "A maximization technique occurring in the statistical analysis of probabilistic function of Markov chains," in *Ann. Math. Stat.*, vol. 41, no. 1, pp. 164-171, 1970

[84] L. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," in *Inequalities*, 3, pp. 1-8, 1972

[85] J. Baker, "The Dragon system-An overview," in *IEEE Trans. Acoustic, Speech, Signal Proc.*, pp. 24-29, 1975

[86] K. Jelinek, "Continuous speech recognition by statistical methods," in *Proc. IEEE*, vol. 64, pp. 532-536, 1976

[87] L. Rabiner, B. Juang, *Fundamentals of speech Recognition*. Englewood Cliffs, New Jersey: PTR Prentice Hall, 1993

[88] G. Forney, "The Viterbi algorithm," in *Proc. IEEE*, vol. 61, pp. 268 – 278, 1973

[89] J. Markel, A. Gray, *Linear Prediction of Speech*. Springer-Verlag, 1976

[90] J. Picone, "Signal modeling techniques in speech recognition" in *Proc. IEEE*, vol. 81, no. 9, 1993

[92] B. Atal, S. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," Journal of Acoustic Society of America, vol. 50, no. 2, pp. 637-655, 1971

[93] S. Srivastava, "Fundamentals of linear prediction," The lecture: Mississipi Satte University Elec.Eng., 1999

[94] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journ. Acoustic Soc. America*, vol. 81, no. 4, pp. 1738 – 1752, 1990

[95] P. Woodland, M. Gales, D. Pye, S. Young, "The development of the 1996 HTK broadcast news transcription system," this is a report on research work conducted by the CU-HTK group.

[96] P. Woodland, G. Evermann, "CU-HTK March 2001 Hub5 system," in *Hub5 Workshop*, 2001

[97] P. Woodland, G. Evermann, "CU-HTK April 2002 switch board system," in *Rich Transcription Workshop*, 2002

[98] B. PEskin, "Improvement of conversational telephone speech," in *Proceedings of ICASSP'99*, 1999

[99] S. Johnson, "Speaker tracking," Master's thesis, MPhil Thesis, Department of Engineering, Cambridge University, UK, 1997

[100] J. Allan, J. P. Callan, M. Sanderson, J. Xu, S. Wegmann, "INQUERY and TREC-7," in Text Retrieval Conference, pp. 148 – 163, 1998

[101] A. Oppenheim, R. Schafer, *Digital Signal Processing*. Englewood Cliff, New Jersey, USA: Prentice-Hall, 1975

[102] P. Zegers, "Speech recognition using neural network," Master's thesis, University of Arizona, 1998

[103] S. Euler, J. Zinke, "The influence of speech coding algorithms on automatic speech recognition," in *Proc. ICASSP*, vol. 1, pp. 621-624, 1994

[104] J. Huerta, R. Stern, "Speech recognition from GSM codec parameters," in ICSLP-98, 1998

[105] J. Huerta, R. Stern, "Distortion-class modeling for robust speech recognition under GSM RPE-LTP coding," *Speech Communication*, vol. 34, pp. 213-225, 2001

[106] S. Gupta, F. Soong, R. Haimi-Cohen, "High accuracy connected digit recognition for mobile applications," in I*CASSP'96*, 1996

[107] T. Soulas, C. Mokbel, D. Jouvet, J. Monne, "Adapting pstn recognition models to the GSM environment by using spectral transformation," in *Proc. ICASSP'97*, 1997

[108] L. Karray, A. Jelloun, C. Mokbel, "Solutions for robust recognition over the GSM cellular network," in *Proc. ICASSP'98*, Seattle, USA, pp. 261-264, 1998

[109] H. Chang, "Is ASR ready for wireless primetime: Measuring the core technology for selected applications," *Speech Communication*, vol. 31, pp. 293-307, 2000

[110] C. Mokbel, L. Mauuary, L. Karray, D. Jouvet, J. Monne, J. Simonin, K. Bartkova, "Toward improving ASR robustness for PSN and GSM telephone application," *Speech Communication*, vol. 23, pp. 141-159, 1997

[111] D. Anderson. (2001) History of speech recognition. [Online]. Available: http://www.netbytel.com/literature/e-gram/technical3.htm

[112] R. Cole, V. Zue, *Survey of the Art in Human Language Technology*. Cambridge University Press, 1998, ch.1: Spoken Language Input

[113] Dragon Naturally Speaking. [On-line]. Available: http://www.astrtech.com/warehouse.html [2004, November 11]

[114] Vodacom South Africa [On-line] Available: http://www.vodaworld.co.za/showarticle.asp?id=420 [2004, November 19]