Using a low-bit rate Speech Enhancement Variable Post-filter as a Speech Recognition System Pre-filter to Improve Robustness to GSM Speech.

Nkululeko S. Mahlanyane

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

November 2003

AMENDMENTS

The following are amendments made to this thesis after its review by both the internal and external markers:

- The thesis title has been changed from "Using a low-bit Speech Enhancement Adaptive Post-filter as a Speech Recognition System Pre-filter to Improve Robustness to GSM Speech" to "The use of a low-bit rate Speech Enhancement Variable Post-filter as a Speech Recognition System Pre-filter to Improve Robustness to GSM Speech." The new title emphasizes the use of the filter as a pre-filter as opposed to its conventional use in telecommunications as a low-bit rate speech enhancement post-filter. The filter is now refered to as variable instead of adaptive. This takes away the possible misunderstanding that the filter characteristics automatically adapt with characteristics of a speech signals.

- pg 18: The sentence in which the statement "the way the information from the nerve fibers is interpreted in the brain" was modified to "Of course, the way the pulses from the nerve fibers are interpreted in the brain needs to be known also and would form part of the recognition process; unfortunately this is not known for certain yet." This clarifies the fact that the EIH model does not attempt to do the interpretation of nerve fiber pulses the same way the brain does it.

- pg 44 and 69: The concept of the error weighting filter is better explained. Its effects on the SNR at different frequencies is clearly stated. The formal derivation of the error weighting filter using equations is included.

- pg 47: The definition of an optimum solution is clearly stated in context.

- pg 51: More information about the HTK toolkit is provided. This added information indicates that the speech recognition systems were not coded from scratch but that

HTK tools were used. Also, the phrase " from scratch" removed so as to make it clear that the recognition systems were not coded by the candidate. In beginning of Chapter 6 the source of the CMS and RASTA source codes is provided. It is also clearly indicated at the beginning of Chapter 7 that the candidate wrote the code for filters experimented with from scratch using C programming language.

- pg 53-54: More information is provided about the Vocal Tract Length Normalization used.

- pg 54 : The independence of the GSM testing data from the training process is clearly explained. It is clearly stated, as it was done experimentally, that the GSM testing data was not used to determine the VTLN warping factor, only clean testing data was used. The practicality of side-based CMS as implemented in this work is discussed and possible alternatives to it in practical situations are mentioned.

- pg 55: The fact that all recognition systems were phone-based continuous systems and not isolated word systems is made clearer by making mention of it in Section 5.1. The use of phone based systems is also motivated in sub-section 5.3.1 in which the operation of a phone-based continuous speech recognition process is discussed.

- pg 56: The concept of state tying is more clearly explained in sub-section 5.3.3.

- pg 60: Equation 6.3 has been corrected together with the context in which it appears.

- pg 73: Equation 7.7 is left as it is and reference to its derivation, as motivation for its correctness, is given.

- Ch 7: Discussions of the statistical significance of results obtained is included. Section 5.5 has been added to describe the matched pairs test that was employed to test the statistical significance of results obtained.

- Ch 7: Reasons for better performance achieved at higher mixture numbers are discussed in greater detail in sub-section 7.41. References to other sections of the thesis that help understand this phenomenon better are given.

- Section 3.2 has been re-written so that it's easier to understand.

- To emphasize the importance of paragraph two in the introduction of Chapter 7, the paragraph has been put in a special separate section (Section 7.1) titled Motivation for Experiments.

- Section 1.3 has been extended to include more background work related to this work. More references we added as a result. Also, the relationship between this work and some of the previous work is described.
- All grammatical errors pointed out have been corrected.
- All the mistakes pointed out in the references have been corrected. The citation style used is the IEEE journal style.

DECLARATION

I declare that this dissertation is my own work. It is being submitted for the Master of Science degree in Electrical Engineering at the University of Cape Town. It has not been submitted before for any degree or examination at this or any other university.

Signature :_____
Nkululeko S. Mahlanyane

Acknowledgments

To my Lord Jesus Christ, thank You for giving me life, the opportunity, the wisdom and the strength to complete this work. I owe you all of my life.

I also want to thank my supervisor Dr. DJ Mashao for his ever present willingness to help whenever I needed his help. Thanks to my fellow students Lerato and Limpho who were always there to help whenever they could. Thanks to my family and close friends for their added support in all that I do, especially Boitumelo. Thank you for believing in me.

To ESKOM, thanks for financial support throughout all my years of studying at the university.

Abstract

Performance of speech recognition systems degrades when they are used to recognize speech that has been transmitted through GSM (Global System for Mobile Communications) voice communication channels (GSM speech). This degradation is mainly due to GSM speech coding and GSM channel noise on speech signals transmitted through the network. This poor recognition of GSM channel speech limits the use of speech recognition applications over GSM networks. If speech recognition technology is to be used unlimitedly over GSM networks recognition accuracy of GSM channel speech has to be improved.

Different channel normalization techniques have been developed in an attempt to improve recognition accuracy of voice channel modified speech in general (not specifically for GSM channel speech). These techniques can be classified into three broad categories, namely, model modification, signal pre-processing and feature processing techniques. Model modifying methods attempt to reduce effects of distortions arising from the channel during the recognition process. On the other hand, signal pre-processing and feature processing techniques attempt to reduce these distortions prior to the recognition process. In most cases techniques from each of these three techniques are used together in one recognition system.

In this work, as a contribution toward improving the robustness of speech recognition systems to GSM speech, the use of a low-bit speech enhancement post-filter as a speech recognition system pre-filter is proposed. This filter is to be used in recognition systems in combination with channel normalization techniques. The filter enhances speech recognition robustness to GSM speech by reducing the effects of GSM speech coding on GSM speech signals. Therefore, in combination with channel normalization, higher levels of robustness can be expected compared to the cases where only channel normalization is used. Cepstral Mean Subtraction (CMS) and RASTA channel normalization techniques are compared for effectiveness in a GSM speech recognition task. CMS was found to be the most effective of the two and was therefore tested in combination with the pre-filter under investigation.

Experiments conducted proved that the proposed use of the pre-filter does improve recognition performance further by about 6% when used in combination with CMS channel normalization.

Contents

List of Figures

List of Abbreviations

ASR:	Automatic Speech Recognition
LPC:	Linear Predictive Coding
PLP:	Perceptual Linear Prediction
MFCC:	Mel-Frequency Cepstral Coefficients
MF-PLP:	Mel-Frequency Perceptual Linear Prediction
EIH:	Ensemble Interval Histogram
GSM:	Global System for Mobile Communications
RPE-LTP:	Regular Pulse Excitation with Long Term Predictor
CELP:	Code-book Excited Linear Prediction
MPE:	Multi-Pulse Excited Linear Prediction
RASTA:	Relative Spectra
CMS:	Cepstral Mean Subtraction
VTLN:	Vocal Tract Length Normalization
HTK:	Hidden Markov ToolKit
WER:	Word Error Rate
PSTN:	Public Switched Telephone Network

Chapter 1

Introduction

Speaking is the most natural way of communication to humans. It is obvious, therefore, that if machines could "hear" and "understand" speech, interaction with them would be much easier. This is the main motivation behind Automatic Speech Recognition (ASR) as a field of research, to enable machines to "hear" and "understand" human speech. Based on its motivation, Automatic Speech Recognition may be defined as a process, carried out by a machine, of extracting information contained in a captured acoustic speech signal and converting it into words.

Research in the field of speech recognition started as long ago as the late 1940s [3]. However significant progress and notable breakthroughs in the field were only made in the last two decades. The first dictation system ever to be made available to the consumer market was only released in 1995 by Dragon Systems with IBM and Kurzweil following a few months later [3]. Today we have a numerous of commercial speech recognition products such as Dragon Naturally Speaking offered by Scansoft and IBM's ViaVoice.

Even with the widespread commercialization of speech recognition products the technology still has a number of shortcomings to overcome. As S. Sandhu in [4] put it, "The ultimate goal of speech recognition technology is to correctly recognize everything spoken by any person in any acoustic conditions, in the minimum possible time with the least cost." This goal has not yet been fully achieved. One of the main challenges yet to be overcome is that of varying performance of speech recognition systems with acoustic environments of use. Environment in this case refers to any medium, characterized by its acoustic properties, through which a speech signal travels both 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 known as a mismatch in training

and testing conditions. Even though these shortcomings have not yet been fully overcome, speech technology has advanced far enough to be applied in practical applications hence the current plethora of commercial speech recognition applications. Some of these speech recognition applications are merely for communicating with machines locally via close-talking microphones (e.g. desktop dictation software). Others access remote recognition engines over certain communications networks. The latter group of applications is currently made up mainly of what has been termed *over-the-telephone automated speech recognition* applications. Over-the-telephone ASR is a special case of automatic speech recognition in which a speech signal is captured at one point of a telephone network, by a telephone device, and is recognized at another point on either the same network or another connecting telephone network. The main telecommunications network used by these applications currently are Public Switched Telephone Networks (PSTN). Though not yet completely perfected, over-the-telephone ASR on PSTN networks has advanced far enough to meet the demands of practical and marketable applications. PSTN network speech recognition applications include call center applications, voice portals, carrier systems and enterprise systems.

The same applications as PSTN speech recognition applications may be developed for mobile phone networks such as GSM (Global System for Mobile Communications) networks. However, such a technology has been implemented to a very limited extent. The reason for this is the fact that GSM networks present challenges that are never encountered in the case of PSTN networks. As mentioned above, one of the main challenges in speech recognition technology is to attain recognition performance that is unaffected by acoustic conditions of use. GSM networks as it stands present an acoustic environment (GSM voice communication channel) that produces very poor speech recognition performance. This work is aimed at reducing the effects of the GSM channel on the quality of speech signals so as to improve recognition rates of such speech by speech recognition system. There are three possible modalities for using speech recognition applications over GSM networks; these are discussed in the next Section.

1.1 Possible Modalities of GSM Network ASR

The three possible modalities (modes of operation) for ASR applications over a GSM mobile network are better known as *Terminal Speech Recognition* , *Network Speech Recognition* and *Distributed Speech Recognition.* Each of these is distinguished from another by the

point on the GSM network at which speech recognition takes place.

1.1.1 Terminal Speech Recognition

In Terminal Speech Recognition (TSR) the recognition process 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.

1.1.2 Distributed Speech Recognition

As its name suggests, Distributed Speech Recognition (DSR) distributes the speech recognition process across different points on the network. One such solution is to have a terminal or mobile device carry out feature extraction process on speech signals and transmit extracted features over the network, possibly through an error-protected data channel, to a remote recognition engine. This approach rules out the effects of the channel on speech signals and exploits the resources of a remote server. One implication of this modality is that DSP capabilities of mobile devices will have to be improved to enable them to perform feature extraction. This factor has cost implications on mobile devices. This is the main disadvantage of DSR modality.

1.1.3 Network Speech Recognition

Network Speech Recognition (NSR) is similar to over-the-telephone ASR modality. In this case, a speech signal propagates through a mobile network to a remote recognition engine. The main disadvantage of this modality is that the speech signal as it travels through the network is exposed to channel imperfections such as transmission noise, data dropouts and interference noise. Its main advantage, however, is that it allows for big machines in remote servers to perform sophisticated recognition operations that cannot be performed on the mobile device like a cell phone.

In this work, a NSR modality is adopted. As mentioned, the disadvantage of this modality is the fact that the speech signal is negatively affected by the channel as it propagates through the network. This work is an attempt to reduce these negative effects with the aim of improving robustness of recognition systems to GSM speech. The two major sources

of degradation of GSM speech that lead to poor recognition performance are mentioned and discussed in the following Section.

1.2 Sources of Degradation of GSM Speech

The main deliberate factor that causes poor performance of speech recognition systems on GSM speech is the speech coding scheme employed in the GSM standard (The Regular Pulse Excited with Long Term Prediction (RPE-LTP)). Speech coding in telecommunications is employed to reduce the bit rate of speech signals so as to use bandwidth available for transmission efficiently. Generally, the quality of speech signals is degraded by speech coding to a certain extent and the recognition thereof by machines becomes poorer. The reduction in bit rate itself degrades performance of ASR systems but on top of that certain coding schemes (especially low-bit speech coding schemes) introduce distortions to speech signals thereby lowering the signal quality even further. This is the case with the speech coding scheme used in GSM. Previous research has shown that generally performance of ASR systems drops with dropping bit rates [5]. One of the conclusions drawn by Lilly and Paliwal in [5] is that bit rates above 16kbps display good recognition performance. In this case PSTN networks do not affect ASR performance much in terms of speech coding because their bit rates are above 16kbps. On the other hand, the GSM full-rate speech codec operates at 13kbps and the half-rate codec at 5.6kbps. The lower bit rate is one of the major reasons why ASR over GSM mobile networks is more difficult to implement than it is over PSTN networks. In telecommunications, filtering methods are often employed to curb the problem of low perceptual quality of low-bit coded speech.

Another major contributing factor to the degradation of speech signals transmitted over GSM networks (termed GSM speech) is the GSM channel noise. A speech signal that has been transmitted through a voice communication channel is often expressed in an equation as,

$$y(n) = h(t) * s(t) + n(t)$$

where $*$ refers to convolution, $h(t)$ is the transfer function of the channel, $s(t)$ a speech signal being transmitted and $n(t)$ the channel additive noise. The same equation 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. The problem is that these two terms are often not known and they often vary

with time. A number of channel normalization techniques which attempt to alleviate the effects of channel noise on speech signals have been developed. Some of these techniques are mentioned and discussed later in this document.

1.3 Background Work in GSM Speech Recognition

A great amount of research work has gone into speech recognition in GSM environments. The leading research group in this field is AURORA which is ETSI's Working Group focused on Distributed Speech Recognition technology. Apart from the work done by AURORA, journals, conference papers, MSc theses and PhD theses relating to this field of research have been written. Lilly and Paliwal [5] studied the effects of speech coding and tandeming on ASR performance. Euler and Zinke [6] also studied the effects of speech coding on recognition performance and went on to analyze the performance of different cepstral features for recognition. Other research work include an attempt to perform recognition from GSM codec parameters assuming availability of such parameters during recognition [7]. Huerta and Stern [8] also proposed a classification method aimed at reducing the degradation in recognition accuracy introduced by full-rate GSM codec. Gupta in [9] used a two-level cepstral mean subtraction robustness approach to improve robustness of ASR systems to GSM environmental noises. Soulas in [10] approached the problem by adapting PSTN models to the GSM environment using spectral transformation. Karray [11] proposed robust HMM architectures for impairments which are encountered in GSM cellular networks. Chang [12] tested the readiness of the current commercial ASR systems for deployment over mobile networks such as GSM networks. His conclusion was that simple small vocabulary task specific systems are robust enough to mobile communications network effects. However, more complex system with large vocabularies and high perplexities showed very low levels of robustness to mobile communication environments. Work more closely related to the work done for this thesis was an investigation of preprocessing and HMM parameter adaptation techniques to improve ASR robustness for PSN and GSM speech [13]. Preprocessing methods investigated are cepstral normalization, high-pass IIR filtering of cepstral trajectories and blind equalization based on adaptive filtering. This work was done with real GSM and PSN data with ASR systems with 50 words in their vocabularies. In this work a speech enhancement post-filter is investigated as a preprocessing filter for robust GSM speech recognition. As will be seen in Chapter 5, in this work real GSM data is used for testing purposes and ASR systems used have vocabularies of about

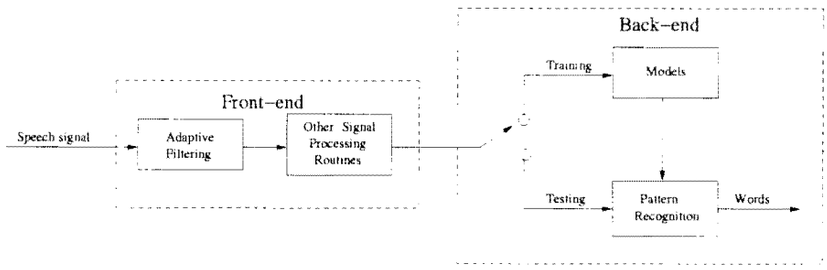


Figure 1.1: Basic Speech Recognition System with a variable pre-filter.

6,000 words.

1.4 Thesis Objectives

In this work, the use of a low-bit speech coding variable post-filter as a speech recognition system pre-filter is investigated as a possible contribution to the solution of the problem of poor recognition performance of GSM speech. The ultimate goal is to improve the robustness of recognition systems to GSM speech. Figure 1.1 depicts a basic recognition system with the variable filter being investigated included in the front-end. The front-end signal processing routines will include a channel normalization technique, e.g. Cepstral Mean Subtraction (CMS). Since the two main causes of poor recognition of GSM speech are channel noise and speech coding, the filter is investigated for use in combination with channel normalization techniques to provide more system robustness. While channel normalization aims to reduce effects of channel imperfections on GSM speech, the variable filter is employed as a way to counter the effects of speech coding. The main question to be answered in this work is, therefore, whether this variable filter does improve robustness of recognition systems to GSM speech; if so, how is this achieved and if not so why not.

The approach taken in this investigative work was to test recognition systems which employed no channel normalization technique with GSM speech to determine their performance on such speech. Different well-known channel normalization techniques where then added to these recognition systems to determine improvements they would bring to performance of these systems. Having determined these improvements, the variable filter is implemented in tandem with the channel normalization technique that proved to be the best among those tested to see if it brought about any further improvements in performance. In particular, a front-end used by the CU-HTK (Cambridge University Hidden

Markov ToolKit) group [14, 15] was used for base-line results and the evaluation of the effectiveness of the variable filter under investigation was measured against these base-line results. This front-end includes a channel normalization technique. It must be noted at this point that no off-the-shelf speech recognition systems were used for experimentation; all systems used were built from scratch using the (Hidden Markov Model ToolKit version 3.2 (HTK v3.2)) following a procedure outlined in Chapter 5.

Another question to answer through this work is whether improvements in perceptual quality (or toll quality) of GSM speech necessarily yield improvements in recognition of such speech. This question stems out from the fact that the filter under investigation is used extensively in practical telecommunication applications. In these applications it is used as a speech enhancement post-filter to improve perceptual quality of low-bit speech coding. This perceptual quality however is judged from the human point of view without any consideration of speech recognition. We seek to know through this investigative work whether improvements in toll quality of speech necessarily brings about improvements in recognition of such speech.

The outline of the rest of this thesis is given in the next Section.

1.5 Thesis Outline

This thesis is divided into two main parts. The first part is made up of theory of speech recognition and the second part of both theory of techniques used and experimental work done. The first part consits of Chapter 2 which is about general speech recognition concepts followed by Chapter 3 which is about the use of HMMs in speech recognition and Chapter 4 which is about Speech Coding theory.

The second part begins with Chapter 5 which describes the experimental setup and procedure followed. This chapter describes how recognition systems tested were built. Also included in this chapter are base-line results against which the effectiveness of the variable filter under investigation will be measured. As mentioned in the objectives of this work above, different channel normalization techniques, in particular cepstral mean normalization [16] and RASTA filtering [17] were tested and compared to determine their performance with GSM speech. Chapter 6 provides theory of the tested channel normalization techniques together with the results of experiments conducted with them. In Chapter 7 the theory of the variable filter is given together with results obtained from experimental work conducted with it. Chapter 8 gives the summary of this work, conclusions drawn

from results obtained from experiments conducted and suggested further work.

Chapter 2

Speech Recognition Concepts

Introduction

Since the advent of automatic speech recognition technology as a field of research many techniques have been devised which aim at making the technology better and better. These range from signal processing techniques to advanced pattern classifiers and statistical models. As may be seen on Figure 1.1, a basic speech recognition system is divisible into two major components known as the front-end and the back-end. The function of the front-end is to perform any necessary signal processing routines on speech signals fed into the system while the back-end involves some method of pattern classification or statistical model training and acoustic pattern recognition (during testing). In this chapter, some of the basic signal processing routines of the front-end and the concept of feature extraction in speech technology are discussed. Well known back-ends are also discussed. The chapter begins with a description of factors used in classifying speech recognition systems. That is followed by front-end and lastly back-end concepts.

2.1 Classification of Speech Recognition Systems

There are different types of recognition systems in use today; these recognition systems are characterized by their designs. Many of them are designed to be task specific. They are often classified according to the following factors [18]:

- **Speaking Mode :** According to the speaking mode recognition systems are classified as either isolated or continuous systems. Isolated systems require users to

speak taking pauses in-between words while continuous systems allow users to speak continuously without any pauses.

- **Vocabulary Size :** Systems' vocabularies differ in size from small (< 10) to large (> 20000).
- **Language Models :** These models are used to put restrictions on sequences of words allowable according to the grammar of the language whose speech the system is designed to recognize. Systems designed for specific tasks often have defined finite task specific grammars.
- **Speaking Style :** Recognition systems range from read speech to natural speech systems.
- **Speaker Enrollment :** Using this classification systems are classified as either speaker dependent or speaker-independent. Speaker dependent systems often need to be trained to adapt them to the user prior to their use while speaker-independent systems are supposed to work equally well with all speakers.
- **Perplexity :** This is a measure of the difficulty of the recognition task in terms of a system's language model and vocabulary size. It is simply defined as the geometric mean of the number of words that can follow a word after the language model has been applied.
- **Transducer :** This has to do with the medium through which speech is captured for recognition. It takes into account factors such as microphones used and channels over which the recognition system is used (e.g. telephone channel).

2.2 Front-ends and Feature Extraction

Speech recognition is not performed directly on digital speech signals fed into a speech recognition system. This is done because there are many sources of variability associated with speech signals. Even two utterances of the same sentence from the same speaker cannot be exactly the same due to these variations. Some of the known sources of these variations are to do with speakers (e.g. different lengths of speakers' vocal tracts), environments of use (e.g. background noise and room reverberation), microphones used for speech capture and pitch of voiced segments of spoken words. Due to these variabilities, it

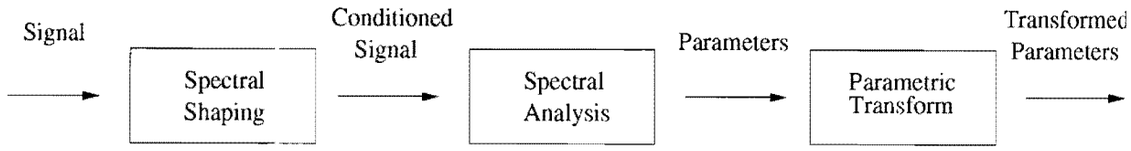


Figure 2.1: Basic front-end routines [After Picone [1]].

is necessary to determine and extract features that are consistently constant in all speech signals of utterances of the same sentence with which one speech sound may be discriminated from another and use them in recognition systems. This is the ultimate goal of signal processing routines in the front-end.

According to the convention used by Picone in [1] fundamental front-end routines can be divided into three parts, namely, spectral shaping, spectral analysis and parametric transformation routines (See Figure 2.1). Spectral shaping involves conditioning a speech signal somehow for optimal spectral analysis.

Spectral analysis is a process of calculating parameters, from spectra of frames of speech, by which one speech sound can be distinguished from another. It is vital to note that there are no standard parameters that spectral analysis methods aim to compute, different methods compute different parameters. For an example, one method may compute total power at different frequencies in a given segment of speech while another computes coefficients of an all-pole model for that segment of speech as parameters. These computed parameters are referred to as features of speech.

Parametric transformation refers to routines that transform parameters calculated from the spectral analysis step into another format that may be more desirable. Cepstral transformation [16] is the most widely used transformation of parameters in speech recognition. It involves the calculation of the cepstrum from parameters computed in the spectral analysis stage. The cepstrum is computed by taking the log of spectral magnitudes and computing the inverse Fourier transform of the log spectrum. The main advantages of cepstral coefficients are the fact that they are decorrelated (which is a vital aspect for Hidden Markov Model (HMM) based systems) and they make features which are more robust to noisy speech and mismatches in training and testing data [19]. Many spectral shaping and parametric transformation methods may be used in any front-end employing any spectral analysis method, they are not spectral analysis method specific even though they may perform better with certain spectral analysis methods than with others.

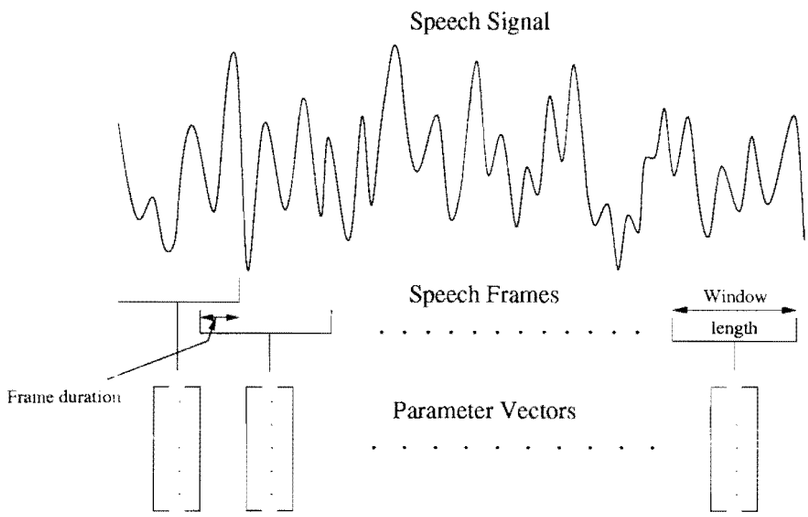


Figure 2.2: Feature extraction illustration.

Feature extraction is a term widely used in speech technology to refer to spectral shaping, spectral analysis and parameter transformation collectively. This term is mainly used to distinguish a spectral analysis method from another since spectral shaping and parameter transformation are often not specific to certain front-ends. This term is adopted and used in the rest of this document.

Feature extraction is performed on short windowed frames of speech, often 20 to 25ms at some frame rate, often 10ms (See Figure 2.2). Even though speech signals are dynamic in nature, these short segments of speech are assumed static. The most popular window function in speech recognition is the Hamming window. The equation of this window function is,

$$w(n) = 0.54 - 0.46\cos(2\pi n/(N-1)) \qquad 0 \leq n \leq N-1 \tag{2.1}$$

where N is the length of the window in samples. Once these features have been computed they 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. It is for this reason that Picone [1] referred to feature extraction as speech signal modeling.

In the following section more is discussed about spectral shaping.

2.3 Spectral Shaping Front-end Routines

As mentioned, spectral shaping signal processing routines in the front-end focus condition a speech signal somehow prior to spectral analysis. The most popular spectral shaping routine is speech recognition is known as pre-emphasis. 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 pre-emphasis coefficient normally in the range of $[-1, -0.4]$ [1]. The main purpose of pre-emphasis is to emphasize high frequencies in a speech signal prior to feature extraction. 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. Pre-emphasis is employed to boost these frequencies. Pre-emphasis is also used in speech coding algorithms [20].

Feature extraction methods are usually classified into three broad categories, namely, Linear Predictive Coding filterbank based and auditory based methods [21]. Each of these classes is discussed in the following sections. Some feature extraction methods employ techniques from more than one of these categories.

2.4 LPC Feature Extraction

LPC modeling was introduced in the early 1970s [22]. Since its advent it has been used, and is still been used in just about any field of speech processing. Some of the fields in which it has found application include speech coding (as discussed in Chapter 4), speech synthesis, speaker recognition and verification, speech recognition and many other speech related technologies.

The concept behind linear predictive coding is to model each sample in a given speech signal (or segment of a speech signal) as a linear combination of a set number of previous samples on the signal. In a form of an equation this idea is expressed as,

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

where p, known as the LPC model order, denotes the number of previous samples used to model the current or the nth sample. $\{a_k\}$ are coefficients of the linear combination known

as linear prediction coefficients.

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,

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

The inverse of $H(z)$ in Equation 2.5 is known as the LPC spectrum and is used as a vocal-tract transfer function [19]. The LPC spectrum tracks peaks and valleys of a speech spectrum so that it appears like a smoothed version of the speech spectrum. It is this property of the LPC spectrum that has found LPC modeling such extensive use in speech processing field.

The goal in solving the LPC model is to compute the LPC prediction coefficients so as to minimize the mean square of the error signal in Equation 2.4 above. The optimal solution for a set of LPC coefficients $a = (a_1 a_2 a_3 ... a_p)^T$ such that the mean square error is minimized is found by solving this matrix equation,

$$a = R^{-1} r \tag{2.6}$$

where,

$$r = (\ r_{ss}(1) \quad r_{ss}(2) \quad ... \quad r_{ss}(p)\ )^T \tag{2.7}$$

and,

$$R = \begin{pmatrix} r_{ss}(0) & r_{ss}(1) & ... & r_{ss}(p-1) \\ r_{ss}(1) & r_{ss}(0) & ... & r_{ss}(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_{ss}(p-1) & r_{ss}(p-2) & ... & r_{ss}(0) \end{pmatrix} \tag{2.8}$$

$r_{ss}(n)$ is the autocorrelation sequence of a given speech signal, $s(n)$, expressed as,

$$r_{ss}(j) = \sum_{n=j}^{N-1} s(n)s(n-j) \tag{2.9}$$

where N is the length (in samples) of a window of speech over which the LPC model is computed. The matrix R is Toeplitz which means the Levinson-Durbin algorithm [23, 24] can be used to iteratively solve for LPC coefficients of the model from the above given equations. This solution of the LPC prediction model is known as the autocorrelation method. There are other methods like the covariance and the lattice (harmonic) methods that are used to solve the LPC model coefficients, but the autocorrelation method is the most favored especially in speech recognition due to its computation efficiency and the fact that it always leads to a stable solution to the LPC model [1, 25]. It is these LPC coefficients that are used in speech recognition as features either as they are or somehow transformed. Once computed, the LPC coefficients are put into a feature vector specific to the segment of speech analyzed.

Use of LPC coefficients in speech recognition systems yields poor performance in noisy environments [1]. For this reason LPC coefficients are hardly ever used as they are in current speech recognition systems. LPC coefficients can me transformed into LPC coefficients cepstra. As mentioned earlier, the cepstrum of a signal is computed as a Fourier transform of the log spectrum. However, for the LPC model it can be shown that the cepstra can be computed from LPC coefficients using the recursive equation below [22, 26, 27].

$$c_n = -a_n + \frac{1}{n}\sum_{i=1}^{n-1}(n-i)a_i c_{n-i} \tag{2.10}$$

The LPC spectrum can also be used for calculating filterbank features [1]. In the next section the filterbank feature extraction method is explained.

2.5 Filterbank Feature Extraction

The simplest implementation of filterbank feature extraction is passing a segment of a 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 features. It is this arrangement of band-pass filters in a sequence that lead to the feature extraction approach termed filterbank approach. These band-pass filters are often not linearly spaced on the frequency axis. The motivation behind this non-linear arrangement comes from the fact that the human ear distinguishes frequencies in an audio spectrum non-linearly. Experimental evidence has also proved that designing filter-banks in

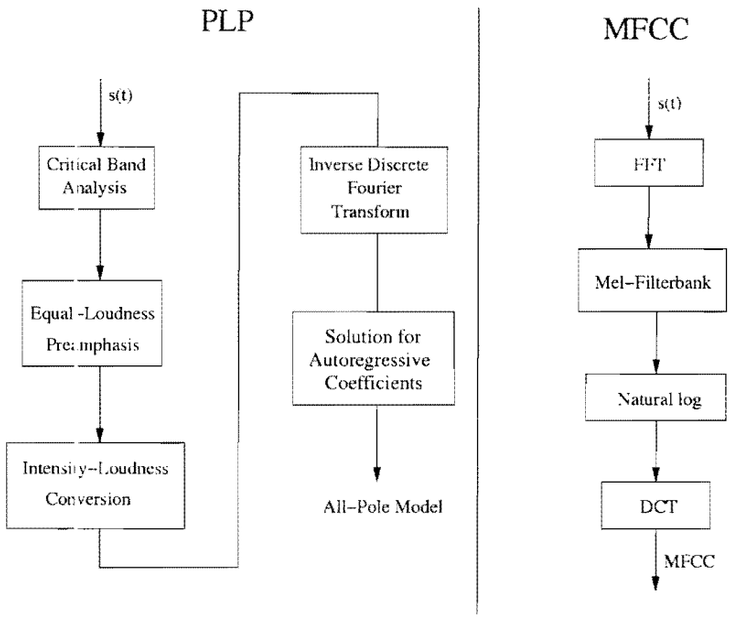


Figure 2.3: Block diagrams of PLP [After Hermansky [2]] and MFCC feature extraction methods.

a similar non-linear manner improves recognition performance [1]. Scales that are normally used to determine the widths, center frequencies, and the spacing of the band-pass filters in a filterbank are the Mel, Bark and the Erb scales.

One of the most popular filterbank based feature extraction methods is the Mel-Frequency-Cepstral-Coefficients (MFCC). The MFCC is currently the most widely used feature extraction method in speech recognition systems. It has been so named because the filter-bank 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,

$$\mathbf{Mel}(f) = 2595 \log_{10}(1 + 1/700) \tag{2.11}$$

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

As may be seen in Figure 2.3, in implementing the MFCC the, magnitude of a spectrum of a window of speech is computed using Fast Fourier transformation. At each frequency the magnitude of this spectrum is multiplied by the magnitude of the filterbank at the

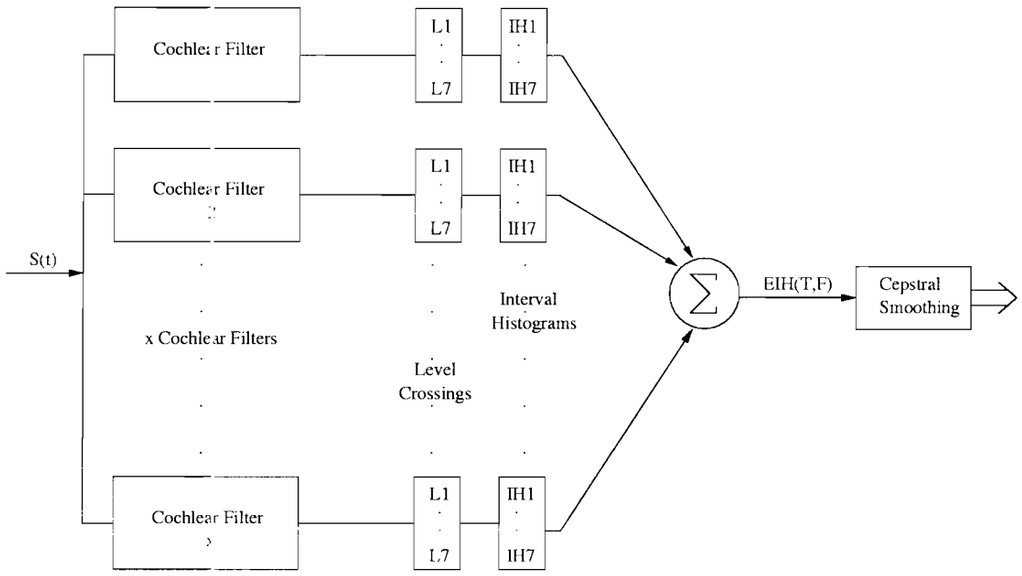


Figure 2.4: EIH model with x cochlear filters and 7 level crossings.

corresponding frequency. All the products from each bandpass filter on the filterbank are summed and these sums are used as extracted features.

Just as in the case of LPC coefficients these features can be transformed somehow. Filterbank 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) in Equation 2.12 below,

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \log(m_j) \cos\left(\frac{\pi i}{N}(j - 0.5)\right) \tag{2.12}$$

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

2.6 Auditory Modeling Feature Extraction

This approach to feature extraction attempts to model the human auditory system (auditory modeling). Most of these methods were developed as attempts to improve recognition performance in noisy environments. The motivation behind them is the fact that humans are able to discriminate between noise and speech in noisy environment; if the human au-

ditory system can be successfully modeled and the models applied to speech recognition, performance in noisy environments would be improved. This approach to feature extraction is also inspired by the amount of knowledge available about the human auditory system. 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 such a feature extraction method is known as the Ensemble Interval Histogram (EIH) [28]. Figure 2.4 depicts an EIH model. This method models short time characteristics of nerve fiber firing rates. This is a sensible approach since auditory nerve fibers are the sole carriers of speech signal information to the brain. If the pulses output by these nerve fibers could be modeled accurately they would give exact features of speech signals that are necessary for interpretation (recognition). Of course, the way the pulses from the nerve fibers are interpreted in the brain needs to be known also and would form part of the recognition process; unfortunately this is not known for certain yet. The nerve fibers don't all fire at the same time; each fiber is activated to fire pulses to the brain by certain sounds. EIH aims to determine those nerve fibers which fire synchronously. An EIH model has three components to it (See Figure 2.4), a bank of cochlear filters, each with an assigned center frequency, equally spaced on the log-scale. Each filter is followed by a set of seven level crossing detectors and the third part is made up of interval histograms one for each level crossing. The bank of filters simulates simulates the cochlear in the human auditory system. The positive going crossing of a level detector simulates a nerve fiber firing. The counts of these crossings are added to histograms of individual frequency bands which are the bandwidths of the cochlear filters. To determine fibers that fire synchronously, interval histograms are all summed to create an ensemble interval histogram. The counts of the ensemble interval histogram are used as features which may be somehow transformed.

Another well known auditory based feature extraction method is an LPC derived Perceptual Linear Prediction (PLP) [2, 29]. As depicted in Figure 2.3, the first stage in PLP analysis is to compute an auditory spectrum from which the LPC model is computed. The 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) spectrum 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 an LPC model is computed. The coefficients of this model are taken as extracted features for the given

segment of 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 in a number of research systems is known as the Mel-Frequency Perceptual Linear Prediction (MF-PLP). As its name suggests, it combines the use of the Mel-filterbank with PLP principles. More detail about this feature extraction method is given in Section 2.7 below.

2.7 MF-PLP

MF-PLP is a feature extraction method that was found to be more robust to mismatches in training and testing conditions than conventional methods like the MFCC and PLP [30]. It was first presented by the Cambridge University Hidden Markov Toolkit (CU-HTK) group in [30]. It was in this work that it was found more robust to noise than normal PLP and MFCC feature extraction methods. The HTK team has since used this front end in virtually all their Switchboard systems (e.g. [14, 15]). These Switchboard systems are large vocabulary continuous speech recognition systems evaluated on telephone conversational speech. MF-PLP has also been used in other systems such as the Dragon Switchboard system [31]. It has been used as a feature extraction method of choice in other systems as well [32, 33]. As will be seen in Chapter 5 this feature extraction method was used to obtain base-line results for this work.

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 the auditory spectrum the LPC model coefficients are computed. These coefficients are taken as extracted features from which cepstra are computed. From this explanation it is clear that 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 the case in normal PLP computation.

2.8 Other Components of Feature Vectors

Often, other components besides extracted features of speech are included in feature vectors. These are usually included because they improve robustness of speech recognition systems. Some of these measures are fundamental frequencies [1], energy measures of given speech segments and time derivatives. Each of these is discussed in the following sub-sections.

Fundamental Frequency

Even though the fundamental frequency is not usually included in feature vectors there are tonal languages for which it has been found necessary to include it (e.g. Chinese) [1]. One of the major problems with the inclusion of the fundamental frequency is the fact it is difficult to estimate reliably. These doubts about it have led to it being not used much in current speech recognition applications.

Energy Measures

The inclusion of the energy measure in the parameter vector in speech recognition is common practice. For a segment of speech N samples long this energy may be computed as [19],

$$E = \log \sum_{n=1}^{N} (w(n)s(n))^2 \tag{2.13}$$

where $w(n)$ is a windowing function. This energy may also be computed without windowing. The log is often taken to simulate the logarithmic response of the human auditory system [1].

Time Derivatives

Addition of time derivatives to parameter vectors has proved to enhance robustness of recognition systems and is virtually standard across all recognition systems. These are time derivatives of features extracted (static features) from a speech signal and any other component that might have been added to the parameter vector (e.g. energy measures). There are three approximations of these derivatives, their formulae with normalization

factors dropped are [34, 35, 36],

$$\frac{d}{dt}s(n) \approx s(n) - s(n-1) \tag{2.14}$$

$$\frac{d}{dt}s(n) \approx s(n+1) - s(n) \tag{2.15}$$

$$\frac{d}{dt}s(n) \approx \sum_{m=-N_d}^{N_d} ms(n+m) \tag{2.16}$$

Another way of expressing Equation 2.16 which is the most commonly used is,

$$\frac{d}{dt}s(n) \approx \sum_{m=1}^{N_d} m(s(n+m) - s(n-m)) \tag{2.17}$$

First order derivatives are computed using one of the above stated approximations and often second order derivatives are computed from first order derivatives using the same approximations. These derivatives are usually normalized. In a normalized manner Equation 2.17 could be [19],

$$\frac{d}{dt}s(n) \approx \frac{\sum_{m=1}^{N_d} m(s(n+m) - s(n-m))}{2\sum_{m=1}^{N_d} m^2} \tag{2.18}$$

2.9 Back-ends

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 acoustic modeling or classification (during training) and acoustic pattern matching (during recognition). The following are some of the well-known pattern classifiers and signal modeling schemes that are either currently being used in speech recognition or have been used before.

Dynamic Time Warping (DTW)

DTW is one of the earliest pattern matching methods used in speech recognition [37]. It takes into account the time varying nature of speech signals and tries to exploit it in pattern matching. In DTW there is a set of stored templates which are often in the form of sequences of feature vectors making up, for example, a word. During recognition, feature vectors of the speech signal to be recognized are matched in time against each of the stored

patterns using some form of a distance measure (e.g. Euclidean distance). The stored pattern that gives the minimum distance is given out as the recognition result of the input speech signal. Since the advent of HMMs, DTW has fallen out of favor for use in speech recognition [38].

Vector Quantization (VQ)

The whole aim of vector quantization is to reduce k-dimensional vectors, R^k, into a finite set of vectors $Y = \{y_i : i = 1, 2, ..., N\}$ [39]. Each y_i is known as a codeword. A complete set of all codewords is called a codebook. The usefulness of VQ is in its dimensionality reduction properties for which it has been applied in compression algorithms such as in image compression algorithms.

Speech recognition using VQ, like DTW, involves storage of templates in the form of codewords in a codebook. During recognition a speech signal is passed through a vector quantizer and the vector outputs are matched against codewords (using distance measures). The recognition result then becomes the codeword that produced a minimum distance.

VQ is not popular as an independent classifier in speech recognition; it is often used in combination with other classifiers and stochastic models such as with HMMs where it is used to reduce the dimensionality of probability distributions of computed feature vectors.

Artificial Neural Networks (ANN)

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 Networks (TDNNs) and Self Organizing Maps (SOM) [40].

One problem which has hindered successful use of NNs in speech recognition is their inability to deal with time warping [40]. This renders them not useful when large time-spans are integrated in systems. It is for this reason that NNs are often used in hybrid systems in combination with other classifiers like HMMs and DTW. In these hybrid systems they are put after the classifiers to manipulate their outputs.

Hidden Markov Models (HMM)

HMMs are the current modeling technique of choice in speech recognition. Since speech recognition systems built in this work are HMM based recognition systems, the next Chapter is dedicated to the theory of HMMs and their application in speech recognition.

2.10 Summary

In this chapter, prominent concepts in speech recognition technology have been discussed. Both the front-end and the back-end concepts of speech recognition systems have been covered. In the next chapter, the theory of HMMs and their use in speech recognition are discussed.

Chapter 3

HMMs In Speech Recognition

Introduction

The theory of Hidden Markov Models (HMMs) was published in the late 1960s and early 1970s by Baum and his colleagues [41, 42, 43, 44, 45]. It was first implemented in speech recognition by Baker [46] at CMU (Carnegie Mellon University), and by Jelinek and his colleagues at IBM in the 1970s [47] . In this chapter, HMMs and their application to the speech recognition problem are briefly explained. The chapter begins with a logical derivation of the mathematical definition of the speech recognition problem. It then outlines the use of HMMs in solving the speech recognition problem. In all equations in this chapter the **bold** style is used to denote vectors.

3.1 Mathematical Definition of the Speech Recognition Problem

The front-end or the signal processing end of the speech recognition system converts an analogue speech signal spoken into a sequence of discrete symbols or vectors (feature vectors) depending on the type of the recognition system. The problem of speech recognition then becomes that of converting these symbols or vectors into a sequence of spoken words (recognition). Assume C to be a sequence of such symbols which are members of set of a finite set of all possible symbols Φ, computed from an utterance made up of a sequence,

W, of words from a vocabulary named Ψ. Mathematically, this is to say,

$$C = c_1, c_2, c_3, ..., c_m \qquad (c_i \in \Phi) \tag{3.1}$$

and

$$W = w_1, w_2, w_3, ..., w_n \quad _{(w_i \in \Psi)} \tag{3.2}$$

The problem speech recognition aims to solve is to produce the most likely sequence of words W given the acoustic observation C. The aim is therefore to maximize the probability of W given C, $P(W|C)$, that is,

$$\widehat{W} = \arg\max_{W} P(W|C) \tag{3.3}$$

Applying Bayes' rule, Equation 3.3 is written as,

$$\widehat{W} = \arg\max_{W} \frac{P(W)P(C|W)}{P(C)} \tag{3.4}$$

Because $P(C)$ is not a function of W and is constant for a given complete sentence, it is eliminated from Equation 3.4 so that the speech recognition problem is mathematically expressed as,

$$\widehat{W} = \arg\max_{W}[P(W)P(C|W)]$$

In most of today's speech recognition systems, the term $P(W)$ is evaluated using a specified statistical language model (e.g. a bigram model) and the term $P(C|W)$ is computed using HMM models. Although discrete symbols were assumed in this derivation, C can be a sequence of vectors. In the next section a formal definition of an HMM is presented.

3.2 HMM Definition

Hidden Markov Models have been derived from Markov Chains. A Markov chain is a statistical model which, at each point in regularly spaced discrete time t, is on state q_t of N possible states. A probabilistic description of such a system calls for the current state to be expressed in terms of all its predecessor states. However, for the first order discrete time case of the Markov chains, the probability of the current state is assumed to depend only on the immediately previous state, i.e,

$$P[q_t = j|q_{t-1} = i, q_{t-2} = k, ...] = P[q_t = j|q_{t-1} = i] \tag{3.5}$$

Considering only those cases where these probabilities are not functions of time results with a state-transition probability defined as $A = a_{ij}$where,

$$a_{ij} = P[q_t = j|q_{t-1} = i] \qquad 1 \leq i, j \leq N \tag{3.6}$$

This set of probabilities obey the following fundamental statistical constraints,

$$a_{ij} \geq 0 \qquad \forall j, i \tag{3.7}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i \tag{3.8}$$

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.

Hidden Markov Models are a special case of Markov chains 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 can only be produced through an underlying set of stochastic processes. A distribution of such probability functions is defined as $B = b_j(k)$ where,

$$b_j(k) = P[\mathbf{o}_t = \mathbf{v}_k|q_t = j], \qquad 1 \leq k \leq M \tag{3.9}$$

M is the number of observable symbols at each state, $\mathbf{o}_t$, the observation made at time t in the process, is $\mathbf{v}_k$, the kth element of the set of all possible observations $V = \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, ..., \mathbf{v}_M$.

Before the HMM process (transition from one state to another for an allocated time units) starts, initial parameters are required. The initial state has to be defined before the process begins. The distribution of this initial state is defined as $\pi = \pi_i$ where,

$$\pi_i = P[q_1 = i], \qquad 1 \leq i \leq N \tag{3.10}$$

From the preceding discussion it is can be deduced that a complete HMM model is defined by the following elements:

1. N, the number of states in the model.
2. M, the number of possible distinct observation in each state.
3. The state transition probability distribution, $A = a_{ij}$.
4. The observation symbol probability distributions,$B = b_j(k)$.
5. Lastly, the initial state distribution $\pi = \pi_i$.

A compact notation used to indicate complete parameters of an HMM model is,

$$\lambda = (A, B, \pi) \tag{3.11}$$

As such, an HMM model is simply a definition of the probability measure for a specified observation sequence $\mathbf{O}$, i.e. $P(\mathbf{O}|\lambda)$.

HMMs pose three problems that need to be solved in order to apply them; these are discussed in the next section.

3.3 Three Problems Posed by HMMs

There are three problems that HMMs present whose solutions make HMMs very useful for a variety of applications. These problems are listed below:

1. **Problem 1**, given the HMM model $\lambda = (A, B, \pi)$ and a particular observation sequence $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, ..., \mathbf{o}_T)$, what is the most efficient way of computing $P(\mathbf{O}|\lambda)$, the probability of the observation sequence given the HMM model. This is known as the evaluation problem.
2. **Problem 2**, given the HMM model $\lambda = (A, B, \pi)$ and a particular observation sequence $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, ..., \mathbf{o}_T)$, how do we work out the state sequence that explains this observation sequence the best. In speech recognition this is known as the recognition or decoding problem.

3. **Problem 3**, involves how we can best compute the parameters of the model $\lambda = (A, B, \pi)$ so as to maximize $P(\mathbf{O}|\lambda)$. In speech recognition this is known as the training problem.

In the following subsections are explanations of some of the well-known and commonly used algorithms to solve these problems.

3.3.1 Solution to Problem 1 - the Evaluation Problem.

The most intuitive solution to the evaluation problem is to calculate the probability $P(\mathbf{O}|\lambda)$ by enumerating every possible state sequence of a given length T. This approach, however, is computationally not feasible. It involves $(2T-1)N^T$ multiplications and N^T-1 additions [21]. An algorithm known as the forward algorithm has been developed for computing this probability more efficiently. This algorithm is explained in the following subsubsection.

3.3.1.1 The Forward Algorithm

The forward variable $\alpha_t(i)$ is defined as the probability of producing the partial observation sequence $\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t$ and state i at the given time t given an HMM model λ. That is,

$$\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t, q_t = i|\lambda) \tag{3.12}$$

It follows from this definition that the product $\alpha_t(i)a_{ij}$ is the joint probability of the partial observation sequence $\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t$ and of state j reached at time $t+1$ from state i at time t. The summation of this product for all possible states i, $1 \leq i \leq N$, in the given model at time t, results with the joint probability of the partial observation sequence $\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t$ and of state j at time $t+1$ from any state i at time t. That is,

$$P(\mathbf{o}_1\mathbf{o}_2...\mathbf{o}_t, q_{t+1} = j) = \sum_{i=1}^{N} \alpha_t(i)a_{ij} \quad 1 \leq t \leq T-1,\ 1 \leq i \leq N \tag{3.13}$$

With state j at time $t+1$ known, the forward variable $\alpha_{t+1}(j)$ is calculated by multiplying this probability (Equation 3.13) by the probability of producing the given observation event $\mathbf{o}_{t+1}$ at state j, $b_j(\mathbf{o}_{t+1})$. That is,

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(\mathbf{o}_{t+1}) \quad 1 \leq t \leq T-1,\ 1 \leq j \leq N \tag{3.14}$$

This equation is computed for every state j in the given model at time t. The computation is iterated for $t = 1, 2, ..., T-1$ i.e. until $\alpha_T(i)$ has been computed. Finally to find $P(\mathbf{O}|\lambda)$ we simply add all computed $\alpha_T(i)$'s. That is,

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.15}$$

The initial value for this algorithm, $\alpha_1(i)$, is set to

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1) \tag{3.16}$$

This computation method involves $N(N+1)(T-1)+N$ multiplications and $N(N-1)(T-1)$ additions which is much more efficient than the direct method [21]. There exists another method very similar to the forward algorithm known as the backward algorithm. This algorithm is briefly explained in the following subsubsection.

3.3.1.2 The Backward Algorithm

Consider the backward variable $\beta_t(i)$ defined as,

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}...\mathbf{o}_T|q_t = i, \lambda) \tag{3.17}$$

This is the total probability of the producing the partial observation sequence $\mathbf{o}_{t+1}\mathbf{o}_{t+2}...\mathbf{o}_T$ (i.e. the rest of the given observation sequence from time $t+1$) given state i at time t of the given model λ. What we are interested in calculating at this point is the probability of being in state i at time t and making a transition to state j at time $t+1$ and producing the rest of the given observation sequence from $\mathbf{o}_{t+1}$ to $\mathbf{o}_T$ for all states j, $1 \leq j \leq N$. Such a probability is expressed as,

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j) \quad t = T-1, T-2, ..., 1, \ 1 \leq i \leq N \tag{3.18}$$

The initial value for this computation is set to,

$$\beta_T(i) = 1, \qquad 1 \leq i \leq N \tag{3.19}$$

This algorithm yields the same results as those of the forward algorithm with the same

computation efficiency. As will be seen later, these two algorithms are together used to solve the training problem of HMMs.

3.3.2 Solution to Problem 2 - the Decoding Problem.

This problem, also known as the decoding problem, says: given the model and an observation sequence, what is the most likely state sequence that produced the observation sequence. The solution to the problem is dependent on the definition of "most likely" state sequence. One possible case is where the "most likely" sequence refers to the most probable state q_t at each time t to produce the observation event $\mathbf{o}_t$ for all observation events in the given observation sequence. Statistically this means we need to maximize the probability of the state sequence $\mathbf{q} = (q_1 q_2 ... q_T)$ given a model λ and an observation $\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 ... \mathbf{o}_T)$. That is, we need to maximize $P(\mathbf{q}|\mathbf{O}, \lambda)$ which is equivalent to maximizing $P(\mathbf{q}, \mathbf{O}|\lambda)$. The well known solution to this problem is known as the Viterbi Algorithm [48, 49].

To explain this algorithm we define the quantity $\delta_t(i)$, as the highest probability along a single path at time t which accounts for t observations and ends at state $q_t = i$, as

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P[q_1, q_2, ..., q_{t-1}, q_t = i, \mathbf{o}_1 \mathbf{o}_2 ... \mathbf{o}_t|\lambda] \tag{3.20}$$

From here it is clear that,

$$\delta_t(j) = [\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij}] \cdot b_j(\mathbf{o}_t) \;\; 1 \leq j \leq N, \;\; 2 \leq t \leq T \tag{3.21}$$

To solve the problem, Equation 3.21 is computed recursively starting from the initial value,

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \leq i \leq N \tag{3.22}$$

until the termination condition,

$$P = \max_{1 \leq i \leq N} [\delta_T(i)] \tag{3.23}$$

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

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T, \;\; 1 \leq j \leq N \tag{3.24}$$

with the initial value set to,

$$\psi_1(i) = 0 \tag{3.25}$$

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

$$q_t = \psi_{t+1}(q_{t+1}), \qquad t = T-1,\ T-2, ..., 1 \tag{3.26}$$

In speech recognition systems this algorithm is used in the recognition process (though not solely). The acoustic vectors computed from the speech signal to be recognized serve 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.

3.3.3 Solution to Problem 3 - the Training Problem.

This 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 $P(\mathbf{O}|\lambda)$. The method used to solve this problem is known as the Baum-Welch method [41, 42, 43, 44, 45] named after L.E. Baum who together with his colleagues was instrumental in the development of the method. This algorithm is also well-known as the forward-backward algorithms because it uses both the forward and backward variables defined in subsection 3.3.1. The explanation of this algorithms starts with the definition of $\xi_t(i,j)$ as

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\mathbf{O}, \lambda) \tag{3.27}$$

This is the probability of being in state i at time t and being in state j at time $t+1$, given the observation sequence $\mathbf{O}$ and the model whose parameters are to be adjusted λ. Using the forward and backward variables this probability can be written as,

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}$$

See [21] for the step by step derivation of this equation.

Let's define $\gamma_t(i)$ as,

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda) \tag{3.28}$$

This is the probability of being in state i at time t given the model λ and the observation sequence $\mathbf{O}$. The summation of this probability over all time units excluding $t = T$ gives the expected number of transitions from state i to any other state, i.e. $\sum_{t=1}^{T-1} \gamma_t(i)$. The summation of $\xi_t(i,j)$ over time (also excluding $t = T$) gives the expected number of transitions from state i to a specified state j, i.e. $\sum_{t=1}^{T-1} \gamma_t(i)$. Expressed in terms of these summations the re-estimation formulae of HMM parameters are,

$$\bar{\pi}_j = \gamma_1(i) \tag{3.29}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.30}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \quad s.t. \mathbf{o}_t = \mathbf{v}_k \tag{3.31}$$

The training procedure using this algorithm can be summarized in the following steps:

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

It can be shown that unless $\bar{\lambda} = \lambda$ (re-estimation iteration termination condition) $P(\mathbf{O}|\bar{\lambda}) > P(\mathbf{O}|\lambda)$.

In speech recognition, these formulae are used to reestimate (iteratively update and improve) parameters of the given model from the given training data.

3.4 Continuous Observation Density HMMs

The discussions above up to the last section refer to the case of HMMs used with discrete observations. Each observation is a member of a defined set of M possible observations.

However, for many applications the observation set is not discrete but continuous or is made up of vectors. Continuous speech recognition is one such application of HMMs in which the observation set is not discrete. This is the motivation behind a special type of HMMs known as continuous observation density HMMs. In this type of HMMs the discrete observation density $b_j(k)$ is replaced by a continuous observation density $b_j(\mathbf{x})$ where $\mathbf{x}$ is a vector. There are usually more than one of these observation densities per state. The observation density of a finite number of mixtures is therefore expressed as,

$$b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk}\chi(\mathbf{x}, \mu_{jk}, \mathrm{U}_{jk}), \qquad 1 \leq j \leq N \tag{3.32}$$

where c_{jk} is the observation density coefficient for the kth observation density in state j. 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,

$$\begin{aligned} \sum_{k=1}^{M} c_{jk} &= 1, \qquad 1 \leq j \leq N \\ c_{jk} &\geq 0, \; 1 \leq j \leq N, \; 1 \leq k \leq M \end{aligned} \tag{3.33}$$

In speech recognition the observation vectors are made up of features 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.

3.5 HMMs in Speech Recognition

To use HMMs in speech recognition, speech signals are represented as a sequence of symbols (e.g. words or phones). In Isolated speech recognition these symbols are complete words whereas in continuous speech recognition they are phones. For each one of these symbols an HMM model is defined. The parameters of these HMMs are observation densities and transition probability distributions.

Since all the recognition systems built in this work are continuous speech recognition systems the rest of this chapter focuses 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. The following subsections briefly describes the recognition network constructed, the training of models in the network as well as how the network is used during the recognition process.

3.5.1 HMM Recognition Network

In a continuous speech recognition system a recognition network that can be viewed at three levels once it's 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. The dictionary lists all the words in the training data set 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 which are either HMM model instances or word-ends.

For small task recognition systems, these networks are usually constructed according to a defined *task grammar* which clearly defines all allowable word sequences. However, for medium vocabulary and large vocabulary (e.g. desktop dictation systems) it is not possible to define such a grammar. In such cases a *word loop* is used which puts all the 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). All the recognition systems built in this work 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.

3.5.2 HMM Training for Speech Recognition

The front-end or the signal processing end of a recognizer extracts feature vectors from each of the given utterances in the training data set. These features are considered as observation vectors in the training of HMMs. Utterances used for training must be segmented and labeled properly according to the phones set used. The sequence of feature vectors for each utterance in the training set are used to train all HMM models of phones appearing in the labeling of that utterance.

3.5.3 The Recognition Process

The recognition problem is that of calculating the most likely word sequence from the given sequence of features using the recognition network. The first stage in the recognition process involves the extraction of features from the utterance to be recognized. These features serve as observations and they are used to compute the most likely state sequence to have produced them from the recognition network. This state sequence is then mapped to the phones sequence then to the word sequence.

Recognition is done by searching for the best path in the recognition network from the start to the end. If an unknown utterance to be recognized has T frames or feature vectors, every path in the network with exactly T emitting HMM states is a potential output. Of these paths the one with the highest probability is given out as the recognition result. The search for the best path 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 [19, 50].

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 an emitting (last state on the HMM model) state 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.

3.6 Summary

In this chapter the mathematical definition of the speech recognition problem was given. The definition of HMMs, the problems they pose along with their solutions were also outlined. The use of HMMs to solve the speech recognition problem was also discussed. In the next chapter, principles of speech coding with more emphasis on the GSM speech codec are discussed.

Chapter 4

Speech Coding

Introduction

Even though speech coding is not the only source of degradation in performance of ASR systems when used to recognize GSM speech, it is the main deliberate factor. Most of the rest of the factors, if not all, come as errors in transmission. Since the method proposed in this work for improving robustness of recognition systems to GSM speech deals with the effects of speech coding, this chapter is dedicated to speech coding algorithms with more emphasis put on the GSM speech coding algorithm, the RPE-LTP (Regular Pulse Excitation with Long Term Prediction) [51, 52].

Speech coding in simple terms is translating the representation of information contained in a speech signal from one digital format into another. For an example, information may be in the form of a simple digitized speech waveform or in the form of coefficients of a filter to be used by a decoder the speech waveform. In this example, translating a digitized speech waveform into coefficients of a filter such that the waveform can be reconstructed is speech coding. The main objective of speech coding is to lower the bit rate of a digitized speech signal to lowest possible bit rates while keeping its perceptual quality good enough.

In telecommunications, speech coding allows for a more efficient use of available bandwidth in the sense that the lower the bit rate the more communication channels can be fitted within a certain limited bandwidth. Since low bit rates mean low bit counts of information, speech coding also allows for a more efficient use of available storage space.

There are three broad classes of speech codecs (speech coding-decoding algorithms), namely, waveform, source and hybrid codecs. In this chapter, commonly used codecs in each of

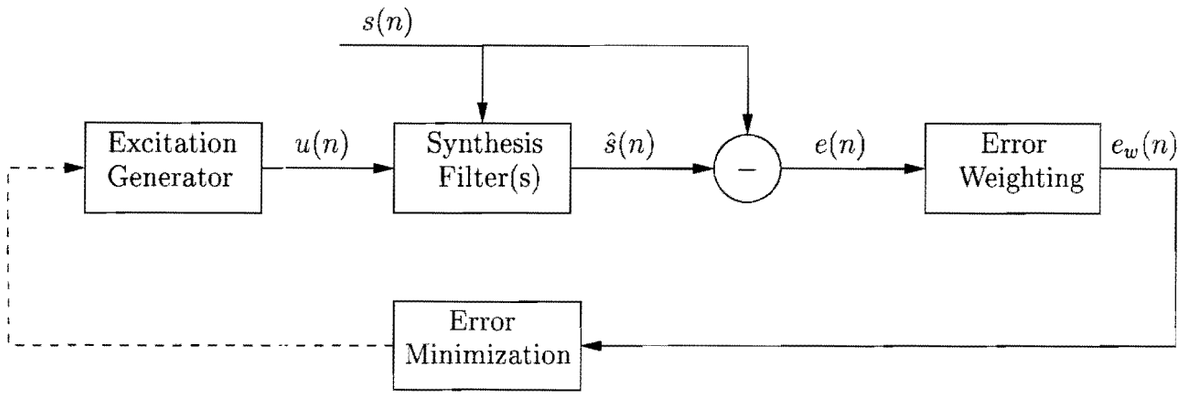
these classes are mentioned and discussed with special emphasis on Hybrid codecs, especially the RPE-LTP codec. One class of Hybrid codecs called Analysis-by-Synthesis (AbS) Predictive Coding is discussed leading to a detailed discussion of the RPE-LTP coding algorithm. Lastly we look at the possible effects of this codec on speech recognition results. Most of the material in this chapter is obtained from [20].

4.1 Wave-form Codecs

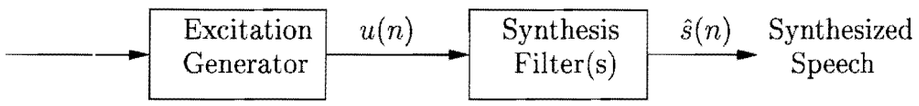
Waveform codecs simply digitize a speech signal without considering its source or how it was produced. They work to digitize any signal, speech or non-speech, i.e. they are source independent. They are the simplest, they require moderate computation power, they operate at high bit rates i.e. 16kbps and above and they have high robustness to transmission errors. Their main disadvantage is their bandwidth inefficiency due to their high bit rates [20]. Examples of waveform codecs are the simple Pulse Code Modulation (PCM), Differential Pulse Code Modulation (DPCM), Delta Modulation (DM) and the Embedded Delta Modulation (EDM).

4.2 Source Codecs

Source codecs take advantage of redundancy in speech signals to predict the speech by modeling its source (i.e. the vocal tract and the vocal cord vibrations) using digital filters. These codecs, also known as vocoders, are specialized for speech and would not work with other signals. Their main advantage is that they operate at low bit rates (as low as 2kbps) [20]. However, they are more sophisticated than waveform codecs and they require above average computation power. The speech quality they produce, although the speech can be perceivable, can be very low and unnatural depending on the bit-rate they operate at. Generally, the lower the bit rate the lower the speech quality. It is for this reason that previous research has proved that the lower the bit rate of speech the lower the recognition performance on such speech [5]. A popular example of vocoders is the LPC vocoder [53].



(a) Encoder



(b) Decoder

Figure 4.1: Block diagrams for Analysis-by-Synthesis encoder and decoder.

4.3 Hybrid Codecs

Hybrid codecs are a combination of waveform and source coding schemes. They are an attempt to develop speech codecs that have advantages of both the waveform and source codecs. They operate at bit rates less than 16kbps and produce good speech quality. Most of them require above average computation power and are very sophisticated.

The popular implementation of hybrid coding is known as Analysis-by-Synthesis (AbS) method [20]. This class of hybrid codecs is of a greater interest in this work than others. The AbS speech coding scheme is discussed in greater detail in the next section.

4.4 Analysis by Synthesis Coding

The general structure of an AbS speech codec is depicted by the block diagram in Figure 4.1. As may be seen in the figure, an AbS codec is made up of three main components, the synthesis filter, the excitation generator and the error weighting and minimization

mechanisms [20]. All AbS codecs share this basic structure; one AbS codec is distinguished from another by the way it generates its excitation sequence.

In the encoder, the synthesis filter produces a synthetic version ($\hat{s}(n)$) of the original signal $s(n)$. Coefficients of the synthesis filters are computed from the original signal, $s(n)$. An error signal ($e(n)$) between these two signals (the original one and the synthetic one) is computed. This error is appropriately weighted and minimized. It is this weighted error ($e_w(n)$) that is sent to the decoder along with the coefficients of the synthesis filter.

The decoder receives a weighted error signal and uses an excitation generator to generate a particular excitation sequence. This excitation sequence models puffs of air which go through the vocal cords from the lungs when one is speaking. The excitation sequence ($u(n)$) fed into the same synthesis filter as the encoder one reproduces the synthetic speech ($\hat{s}(n)$).

Note that this AbS encoder structure forms a closed loop. The advantage of this structure, that makes it superior to open-loop structures, is the fact that the error signal is computed as the difference between the original signal and its synthetic signal output by the decoder.

Each one of the three main components of the AbS codec is discussed separately in the following sections.

4.5 Synthesis Filters

Speech signals show considerable levels of redundancy. This redundancy is in two forms, correlation between samples (in digitized speech signals) as well as the repetition of pitch over a certain number of samples or period (in voiced speech segments). Synthesis filters take advantage of this redundancy to synthesize speech signals using predictive methods. Short Term Predictors (STP) take advantage of sample to sample correlation to predict samples in a signal. In so doing STPs model the short time spectral envelope of a speech signal. Long Term Predictors (LTP) take advantage of pitch redundancy to predict pitch peaks in synthetic speech signals. In so doing LTPs model the fine structure of the speech spectrum. STPs and LTPs are often cascaded in synthesis filters for high quality synthetic speech.

One well known algorithm which is widely used for short term prediction in AbS codecs is the Linear Predictive Coding (LPC) algorithm. In the following sub-section the LPC short term predictor is discussed.

4.5.1 LPC Short-Term Predictor

Coder synthesis filters which use LPC coding are made up of two blocks, namely, the LPC analysis and the LPC synthesis blocks. In the LPC analysis block LPC model coefficients are computed and in the LPC synthesis block a synthetic speech signal is produced using an LPC synthesis filter with LPC coefficients computed in the analysis block. Decoder synthesis filters only have LPC synthesis blocks since they receive LPC coefficients of the synthesis filter from the coder. Both the LPC analysis and synthesis are discussed in this sub-section. To cater for the time varying nature of speech signals, the coefficients of the filters are often computed over short segments of speech, often 10 to 20ms long.

As mentioned in Section 2.4, LPC coding expresses each sample on a speech signal as a linear combination of a set number of previous samples. The synthesized speech segment in a form of an equation, can therefore be expressed as[1],

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k) \tag{4.1}$$

where p is the number of predictor coefficients used called the predictor order. The error is therefore,

$$\begin{aligned} e(n) &= s(n) - \hat{s}(n) && (4.2)\\ &= s(n) - \sum_{k=1}^{p} a_k s(n-k) && (4.3) \end{aligned}$$

and it's z-transform is,

$$E(z) = S(z)A(z) \tag{4.4}$$

where,

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \tag{4.5}$$

Equation 4.5 is the simplest form of the LPC analysis filter. The spectral shape of a speech segmented approximated by the LPC short-term predictor is given by Equation 4.6 (known as the LPC spectrum),

[1]Although Equations 4.1 and 4.2 have been mentioned in Section 2.4 they are mentioned here again for ease of reading.

$$H(z) = \frac{1}{A(z)} \tag{4.6}$$

$$= \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{4.7}$$

$$= \frac{1}{1 - P_s(z)} \tag{4.8}$$

where $P_s(z) = \sum_{k=1}^{p} a_k z^{-k}$ is known as the short term-predictor. Equation 4.8 is the simplest form of the LPC synthesis filter.

The approach employed to compute LPC coefficients for the synthesis filter is to minimize the mean square of the error signal in Equation 4.2. The short time average prediction of this error is defined as,

$$E = \sum_{n=0}^{N+p-1} e^2(n)$$

$$= \sum_{n=0}^{N+p-1} \left[s(n) - \sum_{k=1}^{p} a_k s(n-k) \right]^2 \tag{4.9}$$

where N is the length (in samples) of a speech segment for which the LPC coefficients are computed.

To minimize this error its partial derivative is set to zero, i.e.

$$\frac{\partial E}{\partial a_i} = -2 \sum_{n=0}^{N+p-1} \left\{ \left[s(n) - \sum_{k=1}^{p} a_k s(n-k) \right] s(n-i) \right\} = 0, \quad i = 1, \ldots, p \tag{4.10}$$

With terms rearranged this equation can be written as,

$$\sum_{n=1}^{N} s(n)s(n-i) = \sum_{n=0}^{N+p-1} \sum_{k=1}^{p} a_k s(n-k)s(n-i)$$

$$= \sum_{k=1}^{p} a_k \sum_{n=0}^{N-1} s(n-k)s(n-i) \tag{4.11}$$

If we define,

$$\phi(i,k) = \sum_{n=0}^{N+p-1} s(n-k)s(n-i) \tag{4.12}$$

then equation 4.11 can be written as,

$$\sum_{k=1}^{p} a_k \phi(i,k) = \phi(i,0), \quad i = 1,\ldots,p \tag{4.13}$$

Defining the short-time autocorrelation function as $r_{ss}(j) = \sum_{n=0}^{N-j} s(n)s(n-j)$ it can be shown that $\phi(i,k)$ in Equation 4.12 is a short-time autocorrelation of $s(n-i)$ evaluated for $(i-k)$, i.e.,

$$\phi(i,k) = r_{ss}(i-k) \tag{4.14}$$

Equation 4.12 can therefore be written as,

$$\sum_{k=1}^{p} a_k r_{ss}(|i-k|) = r_{ss}(i), \quad i = 1,\ldots,p$$

The LPC coefficients, $\{a_k\}$, can then be computed using the Levinson-Durbin recursion on Equation 2.6. There are about three methods that can be used to compute these values and they are called the autocorrelation, the covariance and covariance lattice methods [20]. The solution presented above is the autocorrelation method.

LPC synthesis of speech in both the coder and the decoder is done using the synthesis filter in Equation 4.8 once LPC coefficients have been computed. However LPC coefficients do not always guarantee a stable synthesis filter (i.e. the synthesis filter whose pole positions are within the unit circle on the pole-zero plot.). For this reason, LPC coefficients are converted into other formats that guarantee this stability. Often reflection coefficients which are computed as a by-product of the Levinson-Durbin recursion are used in place of LPC coefficients. In the case of using reflection coefficients the synthesis filter is implemented as a lattice structure.

A long term predictor is often used in series with the LPC short term predictor to improve the pitch of the synthetic speech. In the next sub-section the LTP filter is described.

4.5.2 Long-Term Predictor

The short term predictor removes sample-to-sample redundancy but leaves the pitch redundancy of speech untouched. In order to further reduce the bit rate a long term predictor is cascaded to the short term predictor to remove pitch redundancy. It is called the long term predictor because its coefficients are computed from the previous 20 to 160 samples compare to the short-term predictor whose coefficients are computed over 8-15 samples [20]. In the coder synthesis block the LTP filter operates on the output of the short-term predictor whereas in the decoder it operates on the output of the excitation generator. The transfer function of a general long-term predictor is,

$$\begin{aligned} H(z) &= \frac{1}{1 - P_l(z)} \\ &= \frac{1}{1 - \sum_{k=-m_1}^{m_2} G_k z^{-(\alpha+k)}} \end{aligned} \tag{4.15}$$

where,

$$P_l(z) = \sum_{k=-m1}^{m2} G_k z^{-(\alpha+k)} \tag{4.16}$$

is known as the long-term predictor and α is the fundamental pitch period or its multiple. The approach adopted in computing the set of coefficients $\{G_k\}$ and α, is to minimize the mean-squared error between the output of the long term predictor and its input. When $m_1 = m_2 = 0$ the LTP is known as a 1-tap predictor. The output of a 1-tap predictor, $y(n)$, can be expressed as follows in terms of $r(n)$ the output of the LPC analysis filter,

$$y(n) = Gr(n - \alpha) \tag{4.17}$$

The mean squared error is expressed as,

$$\begin{aligned} E &= \sum_{n=0}^{N-1} [r(n) - y(n)]^2 \\ &= \sum_{n=0}^{N-1} [x(n) - Gx(n - \alpha)]^2 \end{aligned} \tag{4.18}$$

To minimize E we set $\frac{\partial E}{\partial G} = 0$ which results with,

$$G = \frac{\sum_{n=0}^{N-1} r(n)r(n-\alpha)}{\sum_{n=0}^{N-1}[r(n-\alpha)]^2} \tag{4.19}$$

Substituting Equation 4.19 into 4.18 results with,

$$E = \sum_{n=0}^{N-1} r^2(n) - \frac{\left[\sum_{n=0}^{N-1} r(n)r(n-\alpha)\right]^2}{\sum_{n=0}^{N-1}[r(n-\alpha)]^2} \tag{4.20}$$

To minimize E we must maximize the second term in equation 4.20. This term is computed for all possible values of α (often in the range 20 - 160 samples) and the value that maximizes the term is taken as the fundamental pitch and is used (or its integer multiple) in Equation 4.19 to determine the LTP filter gain, G. For a K-tap predictor α is computed first and a set of $K \times K$ equations is solved for filter gains $\{G_k\}$.

4.6 Error Weighting

The error weighting filter is needed to reduce the amount of quantization noise associated with speech coding which is perceived in senthesized speech. The filter operates on the error signal in the coder, hence it's been termed error weighting filter. It has been discovered in the theory of auditory masking that perceived noise is partially or totally masked at formant frequency regions where the speech energy is high (More is discussed about the theory of auditory masking in Chapter 7). The strategy to reduce this noise is therefore to attenuate formants of speech signals moderately in order to attenuate noise in those regions too. This attenuation, however, amplifies noise in inter-formant regions. The end results will therefore be reduced SNR at formant frequencies and increased SNR in non-formant frequencies. The degree to which formants are attenuated must therefore be such that the noise in inter-formant regions does not rise above audible levels.

In [54], Atal and Schroeder showed that the quantization noise in a senthisized speech signal is given by,

$$\begin{aligned} |N(z)|^2 &= \left|\hat{S}(z) - S(z)\right|^2 \\ &= |\Delta(z)|^2 \left|\frac{1-F(z)}{1-P_s(z)}\right|^2 \end{aligned} \tag{4.21}$$

where $|\Delta(z)|^2$ is the power spectrum of the noise at the output of the quantizer and $F(z)$ is a feedback filter. When a weighting filter, $W'(z)$, is employed to reduce quantization noise in the synthesized speech, then,

$$\Delta(z) = N(z)W'(z) \tag{4.22}$$

Substituting Equation 4.22 into Equation 4.21 yields,

$$W'(z) = \frac{1 - P_s(z)}{1 - F(z)} = \frac{A(z)}{B(z)} \tag{4.23}$$

In [54] it was found that an appropriate choice for $B(z)$ is $A(z/\gamma)$. Therefore, the general transfer function of the weighting filter used is,

$$\begin{aligned} W'(z) &= \frac{A(z)}{A(z/\gamma)} \\ &= \frac{1 - \sum_{k=1}^{p} a_k z^{-k}}{1 - \sum_{k=1}^{p} \gamma^k a_k z^{-k}} \end{aligned} \tag{4.24}$$

Figure 7.1 shows the spectra of this filter; from the figure it can be seen that the filter does attenuate formant peaks and amplify inter-formant regions. Note that this filter is the LPC analysis filter cascaded to another synthesis filter whose transfer function is,

$$W(z) = \frac{1}{A(z/\gamma)} \tag{4.25}$$

This synthesis filter is known as the weighted synthesis filter. This means the original synthesis filter of Equation 4.8 is simply replaced by the weighted synthesis filter when error weighting is performed. In the next section the excitation generator of an AbS codec is discussed.

4.7 Excitation Generator

The defining component of an AbS coding scheme is an excitation generator. Different AbS coding schemes use different algorithms for determining the optimum excitation sequence. Optimum in this case means the sequence that minimizes the error between the synthesized and original speech signals.

An excitation sequence is defined by the heights $\{\beta_k\}$ and positions $\{m_k\}$of its pulses. It is expressed as follows in the form of an equation,

$$v(n) = \sum_{k=0}^{M-1} \beta_k \delta(n - m_k) \qquad n = 0, \ldots, N-1 \tag{4.26}$$

where M is the number of pulses in an excitation sequence. The input speech to be coded is inverse filtered (analysis) and weight filtered (weighted synthesis) before the error between the original and the synthesized speech is computed.

The weighted synthesized speech is computed by convolving the output of the LTP $u(n)$ with the impulse response of the weighting filter $h(n)$, i.e.,

$$\widehat{s}_w(n) = u(n) * h(n) \tag{4.27}$$

Assuming, without loss of generality, the use of an adaptive codebook LTP, $u(n)$ is expressed as,

$$u(n) = v(n) + Gc_\alpha(n) \tag{4.28}$$

where $v(n)$ is the excitation signal, G the gain factor and $c_\alpha(n)$ the codeword selected from the codebook. Substituting Equation 4.28 into Equation 4.27 gives,

$$\widehat{s}_w(n) = v(n) * h(n) + Gc_\alpha(n) * h(n) + \widehat{s}_0(n) \tag{4.29}$$

where,

$$\widehat{s}_0(n) = \sum_{k=1}^{p} a_k \gamma^k \widehat{s}_0(n-k) \qquad n = 0, \ldots, N-1 \tag{4.30}$$

is the zero-input impulse response of the weighting filter.

Substituting Equation 4.26 into Equation 4.29 yields,

$$\begin{aligned} \widehat{s}_w(n) &= \sum_{i=0}^{n} \left(\sum_{k=0}^{M-1} \beta_k \delta(n - m_k) \right) h(n-i) + Gc_\alpha(n) * h(n) + \widehat{s}_0(n) \\ &= \sum_{k=0}^{M-1} \beta_k h(n - m_k) + Gy_\alpha(n) + \widehat{s}_0(n) \end{aligned} \tag{4.31}$$

where,

$$y_\alpha(n) = c_\alpha(n) * h(n) \tag{4.32}$$

Using Equation 4.31 the error between the original weighted and synthesized weighted speech is,

$$\begin{aligned} e_w(n) &= s_w(n) - \widehat{s}_w(n) \\ &= x(n) - \sum_{k=0}^{M-1} \beta_k h(n - m_k) \end{aligned} \tag{4.33}$$

where,

$$x(n) = s_w(n) - G y_\alpha(n) - \widehat{s}_w(n) \tag{4.34}$$

The mean square weighted error is given by,

$$\begin{aligned} E_w &= \sum_{n=0}^{N-1} e_w^2(n) \\ &= \sum_{n=0}^{N-1} \left[x(n) - \sum_{k=0}^{M-1} \beta_k h(n - m_k) \right]^2 \end{aligned} \tag{4.35}$$

To compute pulse heights and positions this mean square error is minimized by equating its derivative with respect to pulse heights to zero, i.e.,

$$\frac{\partial E_w}{\partial \beta_i} = -2 \sum_{n=0}^{N-1} \left[x(n) - \sum_{k=0}^{M-1} \beta_k h(n - m_k) \right] h(n - m_i) = 0, \qquad i = 0, \ldots, M-1 \tag{4.36}$$

After reordering this equation can be written as,

$$\sum_{n=0}^{N-1} x(n) h(n - m_i) = \sum_{k=0}^{M-1} \beta_k \sum_{n=0}^{N-1} h(n - m_k) h(n - m_i) \tag{4.37}$$

If we define,

$$\phi(i, j) = \sum_{n=0}^{N-1} h(n - i) h(n - j) \tag{4.38}$$

the autocorrelation of $h(n)$ and,

$$\psi(i) = \sum_{n=0}^{N-1} x(n)h(n-j) \tag{4.39}$$

the cross-correlation between $x(n)$ and $h(n)$, Equation 4.37 becomes,

$$\sum_{k=0}^{M-1} \beta_k \phi(m_i, m_j) = \psi(m_i) \qquad i = 0, \ldots, M-1 \tag{4.40}$$

This equation results in M equations with $2M$ unknowns. From this equation a set of pulse heights $\{\beta_k\}$ and positions $\{m_i\}$ which minimize Equation 4.35 are computed and are used as heights and positions of the generated excitation pulse sequence.

Calculating the optimum solution (i.e. the most mathematically accurate values of $\{\beta_k\}$ and $\{m_i\}$ which minimize Equation 4.35 without any assumptions) to Equation 4.40 is computationally expensive; for this reason, algorithms for sub-optimal solutions (less accurate values of $\{\beta_k\}$ and $\{m_i\}$ to minimize Equation 4.35 which are effective and make mathematically correct assumptions) to the equation have been developed. These algorithms are the ones that define and differentiate one AbS coding scheme from another. Some of the well-known of these algorithms are the Codebook Excited Linear Prediction (CELP) algorithm [55], Multi Pulse Excited (MPE) algorithm [56], and the Regular Pulse Excited (RPE) algorithm [51, 52]. The GSM codec employs the RPE-LTP algorithm which is the RPE algorithm with a long term predictor. This algorithm is discussed in the next section.

4.8 The RPE Algorithm

The duty of an excitation generator, given a weighted error signal, is to produce a certain number of pulses with certain heights at certain positions, i.e., solve Equation 4.40. The RPE approach predefines positions of pulses regularly spaced hence the term Regular Pulse Excitation. Since pulse positions are pre-defined the RPE algorithm concentrates mainly on pulse heights. Pulse positions are determine by,

$$m_i^{(k)} = k + iD, \quad k = 0, ..., D-1; \quad i = 0, ..., M-1 \tag{4.41}$$

where k is the position of the first pulse, D the distance between any two pulses on the pulse train and M the number of pulses determined by the integer devision of the LPC frame length N by D. Once these positions have been thus determined the RPE algorithm has to solve the heights of the pulses in these pre-determined positions. The RPE algorithm is very computation intensive. It involves solving Equation 4.40 for every possible excitation sequence $\left\{\beta_i^{(k)}\right\}$. Possible excitation sequences are determined by the initial phase, k. This means Equation 4.40 is solved D times, each time for a different value of k. For every one of these sequences the mean square error of Equation 4.35 is solved and whichever sequence minimizes it is chosen as the excitation sequence for the frame. In total the RPE algorithm requires solving M equations for $2M$ unknowns D times [20].

There is a simplified version of the algorithm which is less computation intensive. This simplification comes from using the autocorrelation formulation [57, 58] which reduces the matrix of Equation 4.40 into a Toeplitz matrix which can be solved efficiently using the Levinson-Durbin recursion. The matrix also becomes independent of the initial pulse position which implies that the matrix is inverted only once in an LPC frame rather than D times in the frame. The total number of operations required to solve Equation 4.40 is reduced to $M(N-D)/2$. Even further simplification is achieved by eliminating the matrix inversion. Details of this simplification can be found in [20].

The history of the GSM standard and the use of the RPE-LTP algorithm in the standard are briefly discussed in the next section.

4.9 The use of RPE-LTP Algorithm as a GSM codec

The history of the GSM standard goes as far back as 1982 when the Nordic Telecom and Netherlands PTT proposed to the Conference of European Post and Telecommunications the development of a digital cellular standard [59]. In 1986 to the beginning of 1987 main GSM radio transmission techniques including the Time Division Multiple Access (TDMA) and the RPE-LPC speech coding scheme were chosen [60, 59]. The first GSM network operator started in 1992 [59]. At the time of writing, there are 793 million GSM users globally and 14 million users in South Africa alone [61].

The GSM standard was decided upon after having run a number of comparative experiments to determine the best among different digital systems which were then proposed. Speech codecs that were tested are the Sub-band Codec with block Adaptive PCM (SBC-APCM), Sub-band Codec with Adaptive Delta PCM (SBC-ADPCM), the Multi-Pulse

Excited LPC codec with Long Term Predictor (MPE-LTP) and the Regular-Pulse Excited LPC codec with Long Term Prediction (RPE-LTP) [60]. The speech codecs were compared in terms of the speech quality, robustness to channel errors, processing delays and computational complexity; the RPE-LTP was found to be the best and was therefore chosen.

4.10 Effects of the RPE-LTP on Recognition Performance

Huerta in [62] proved that the short term residual of the RPE-LTP speech codec produces a significant level of quantization distortion which in turn affects GSM speech recognition performance. This distortion affects different classes of phonemes differently. From Huerta's results in experimenting with these different classes of phones it was reported that the most severely affected class is made up, mostly, of nasals. The mere reduction in bit rate of speech due to speech coding affects recognition performance [5]. Lilly and Paliwal in [5] concluded that speech recognition performance tends to degrade with the decrease in the bit rate of speech signals.

4.11 Summary

In this chapter different speech coding schemes have been discussed with special emphasis on the AbS Hybrid codec. Each of the three main components of an AbS codec, namely, the synthesis filters, error weighting and minimization criteria and excitation generator are discussed. The RPE-LTP excitation generator scheme which is used in the GSM standard has been discussed in greater detail. Lastly the effects of the RPE-LTP on speech recognition performance are mentioned. In the next chapter the experimental setup and procedure followed in this work are discussed. The base-line speech recognition system used for experimentation and the building procedure thereof are described.

Chapter 5

Experimental Setup and Base-line Results

Introduction

In this work, no off-the-shell speech recognition systems were used for experiments. All recognition systems were developed using the Hidden Markov Model Toolkit version 3.2 (HTK v3.2) [63]. HTK is not a ready made speech 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. The process of developing the recognition systems using HTK tools is described in detail in this chapter.

Even though speech recognition systems differ with tasks for which they are developed their development is generally the same. This process of development is experimental in nature. It usually starts off with the construction of what is known as a *base-line recognizer* whose performance is measured in percentage Word Error Rate (% WER) or percentage word recognition accuracy it produces on a given task. From this point on, well educated experiments are conducted to improve the performance of this baseline recognition system until the optimum performance is obtained on the given task (See, for an example, [64, 65]). This might involve fine-tuning certain parameters of the base-line system.

The base-line recognition system for this work and the construction thereof are described in detail. All subsequent recognition systems built for experimentation are of the same

design as the base-line recognition system. Details of the general design for all recognition systems are listed in the next section.

For base-line results, the base-line clean speech recognition system is tested with two types of data, namely, clean speech and the real GSM speech. These two results are compared and comments about them are made.

Although the recognition system design implemented in this work is that of the 1993 HTK tied-state tri-phone system [64], the front-end used is of the CU-HTK (Cambridge University HTK group) April 2002 Switchboard System [15]. This front-end was chosen because it was used in a system that was evaluated with real GSM speech just as it is done in this work. Constituents of this front-end are listed and discussed in Section 5.2.

5.1 General Design

All of the following aspects of recognition systems listed below were applied on each one of the systems built.

- All systems are phone based continuous speech recognition systems. There was a total of 49 phones which we used in the construction recognition systems.
- All systems are HMM recognition systems. All HMM models for all 49 phones used were 5-state, left-to-right non-skipping models with continuous Gaussian mixture densities. The number of mixtures on HMMs was kept as one until later stages of the building process where they were incremented to reach peak performance.
- They were all built using medium vocabulary clean speech TIMIT database training set [66] either as it is or a somehow modified version of it. Recognition systems were evaluated using the TIMIT core test set [66] either as it is or somehow modified.
- The TIMIT dictionary [66] was used with its 61 phones set mapped to CMU's 48 (including silence) phones set listed in [67]. With the addition of the short-pause (sp) the number of phones were in total 49.
- All systems built are tied-state tri-phone systems. Their training procedure is described in Section 5.3.
- Even though different feature extraction methods have been used for different recognition systems built, there are similarities in how feature extraction was performed.

Feature extraction was performed on 25ms long segments of speech. The Hamming window was applied on the short segments of speech.1st and 2nd order derivatives of the static features were computed for all feature extraction methods applied. Pre-emphasis with a coefficient of 0.97 was applied.

- The best language model settings were worked out to be 30 and 50 for language model (LM) probability weighting and word insertion penalty respectively. These language model settings were used in all the experiments performed.

The training procedure of these systems is described in detail in Section 5.3. In the next section components of the base-line front-end are listed and briefly discussed.

5.2 Base-line Front-end

Since speech recognition system development is experimental in nature, front-ends usually differ from system to system. There is therefore no front-end standard across all systems. The chosen front-end for base-line results in this work is the one used by the HTK group in the DARPA'02 evaluations [15]. These evaluations involved testing of speech recognition systems developed by different leading research groups in speech recognition. Among different testing speech data was GSM speech. This is the motivation behind the choice of this front-end. This front-end consists of the following:

- Bandwidth reduced to 125 - 3800
- 12 MF-PLP cepstral features + the zeroth cepstral coefficient (C0) and 1st and 2nd order derivatives
- Side-based cepstral mean and variance normalization
- Vocal tract normalization in both training and testing data

The MF-PLP feature extraction was first used by the HTK group in their 1996 Broadcast News Transcription System and it was found more robust and consistent under mismatched conditions than both the MFCC and the simple PLP [65]. This proved robustness of the MF-PLP from previous research was motivation for its use for base-line recognition results in this work.

In [68, 69] 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 the telephone network. This proved that the effects 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 [68] it was even concluded that in some cases band-limited clean speech recognition systems produced lower word error rates than telephone speech trained recognition systems even when tested with real telephone speech. This is the motivation behind the use of band-limiting in the base-line front-end. The advantage of using band-limiting is also to do with band-stopping any unwanted high and low frequencies.

Side-based cepstral mean and variance normalization is termed side-based because the cepstral mean and variance are computed before the recognition process from a somehow clustered subset of data. In our implementation, side-based cepstral means and variances were computed per speaker for all data types. It is important to note that data were not mixed in computing the cepstral mean vector for CMS. For an example, CMS was performed per speaker for clean speech and GSM speech separately. In practical cases where various instruments are used CMS can be performed for each instrument or environment separately. However, this may be impractical in which case other groupings can be defined. For example, in the case of telephone speech different phone sets have microphones with different characteristics. In such a case it is imposible to do side-based CMS for each microphone but data can be clustered in terms of caller and the called party speech signals in the conversation.

Vocal tract length normalization is a simple speaker normalization technique derived from the fact that different speakers have different vocal tract lengths. 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 warp factor to data and the warp type (i.e. linear, non-linear or piece-wise). In the implementation of VTLN in this work the warp type used is piece-wise defined with a warping factor chosen as the one that gave the best performance for all speakers both in the training and the clean testing data. The GSM testing data was not included in the experimental determination of the VTLN warping factor only the clean testing data was used. This effectively left the GSM testing data completely independent of the training process. This choice is obviously not optimum but it actually brought about significant

improvements in robustness to real GSM data. The warp factor is applied on the mel-filterbank as suggested in [70, 71, 72].

5.3 Training Procedure

The procedure followed in building recognition systems from scratch is described in this section. The recognition system was developed using the same design as that of the HTK 1993 clean speech recognition system [64]. There are three stages over which a complete tied-state recognition system is built and these are:

- Data preparation
- Creating Mono-phone HMMs
- Creating tied-state Tri-phone HMMs

Each of these three stages is made up of a number of sub-stages. These stages together with their sub-stages are discussed in the following Sub-sections.

5.3.1 Data Preparation

Data preparation involves creating the dictionary, language bigram model, recognition network and coding the training data.

The dictionary used is the TIMIT database dictionary [66]. As is, it is not compatible with HTK tools. The format of the database was altered to an HTK compatible format. Also, its 61 phones set was reduced to the 48 phones CMU set by deleting all glottal stops and replacing all closures preceding voiced stops and unvoiced stops by generic voiced and unvoiced closures respectively. Phone label files for each speech file were changed appropriately to reflect the new phone set. The dictionary transcriptions were also changed to the new phone set.

In task specific recognition systems, a task grammar is usually defined and used to construct a recognition network. This makes the recognition system much more accurate. Recognition systems built in this work are not task specific and therefore no definite grammar exists. The grammar is therefore statistically defined from the list of all words available in the vocabulary of the TIMIT database. In this work a bigram language model is used.

The recognition network is constructed as a word loop such that all words are in parallel with one another. This arrangement of the recognition network ensures that every word in the vocabulary can be followed by every other word in the same vocabulary. To improve performance, the bigram model is used to attach transition probabilities on every sequence of words. This means that the probability of one word being followed by another is not the same as every other two word sequence but is determined statistically from the bigram model. The recognition word network is mapped into a model network using word transcriptions in the dictionary. The model network is in turn mapped into a state network by joining HMM models together. This means a complete recognition network is viewed at three levels, the word level, the model level and the state level. During recognition, which takes place at the state level of the network, a sequence of phones is produced which is mapped into words using dictionary word transcriptions. These words are then mapped into sentences (sequences of words) using the bigram language model. This means the systems perform phoneme recognition and not isolated word recognition. Phone recognizers are more effective for continuous speech recognition systems than isolated word recognition systems.

The coding of speech data involves parameterization of speech files according to signal processing specifications in the front-end of the system.

5.3.2 Creating Mono-phone HMMs

In both the constructions of the mono-phone system and of the tied-state system, a 3-state left-to-right single-Gaussian mixture HMM was defined for each of the 48 CMU lexicon monophones [67]. Most of these HMMs are without skips. This has proved to be a good HMM topology for phone-based recognition systems [19]. In essence these HMMs are 5-state models, however, the entry and the exit states are non-emitting. These two non-emitting states are used to join models at the model level of the recognition network.

All models were initialized by repeatedly applying the Viterbi algorithm to find the most likely state sequence corresponding to each training utterance in the training data and updating the HMM parameters. In the process of the Viterbi algorithm the log probability of the training data is computed and the algorithm is applied repeatedly until the log probability cannot be improved anymore (termination condition). After initializing, model parameters are re-estimated using the Baum-Welch algorithm. Once these monophones have been trained a complete mono-phone speech recognition system has been constructed.

Systems built and tested in this work are tied-state tri-phone systems. This means the monophones system had to be developed further into a tied-state tri-phone system.

5.3.3 Creating Tied-state Tri-phone HMMs

Tying of HMMs means making two or more share certain parameters pertaining to HMMs. Some of these parameters are transition matrices, state mixtures, mean vectors, and others. A well educated state tying improves the robustness of parameter estimation during the HMM training process [19]. In recognition systems built for this work, HMMs are tied by allowing phones that fall in the same phonetic group to share parameters, for example, nasals are tied with other nasals. Tied-state HMMs are built by first creating tri-phone HMMs from mono-phone HMMs. This process starts by generating a list of tri-phones using a given list of monophones and phone level transcriptions of training utterances. For every one of the tri-phones in this list of tri-phones a tri-phone HMM is cloned from mono-phone HMMs. These tri-phone models are then tied. There were two tyings performed: one was that of transition matrices of all tri-phones that had the same middle phone and the second one was that of tri-phones that belonged to the same phonetic sets (e.g. Nasals, Glides etc). In this case all these triphones will share all the information that pertain to an HMM model (e.g. variances and means). The second tying is done so that tied tri-phones may share data thereby making more robust parameter estimates.

For all the systems built the number of mixtures per HMM model state were progressively incremented up to 9 in order to determine the number of mixtures where peak performance occurred.

5.4 GSM Files Recording

Due to unavailability of a comprehensive medium vocabulary database of GSM data, recordings of GSM speech files were made over the Vodacom network for testing. Vodacom is a South African GSM network company. In recording, two Mobile Stations (cellular phones) were used, one connected at the transmitting end and the other at the receiving end (See Figure 5.1). The one at the transmitting end was used to transmit speech signals from the TIMIT database core test set [66] over the GSM network to the receiving end. The receiving end received signals from the network and recorded them. The transmitter and the receiver were two computers appropriately networked over the IP

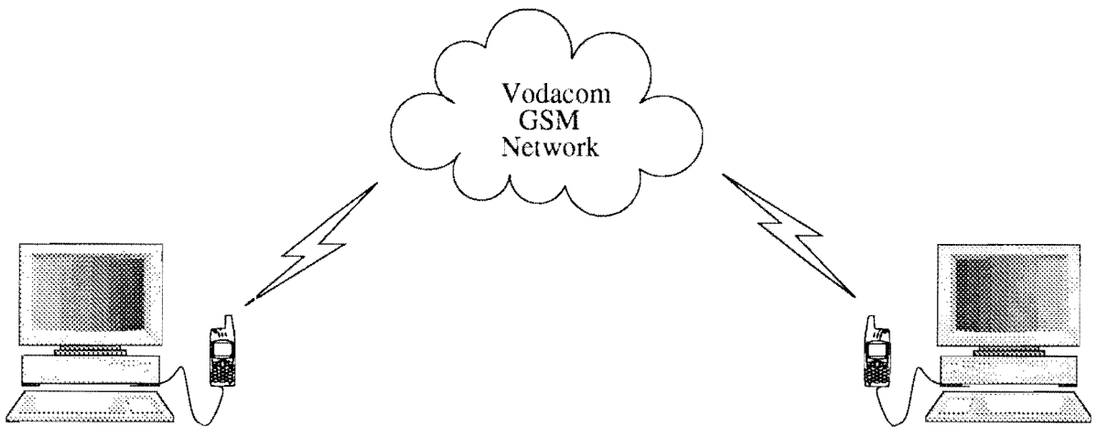


Figure 5.1: GSM files recording setup.

network for synchrony purposes.

These files were recorded at 16kHz. For the purpose of experimentation they were down-sampled to 8kHz. All experiments, both in testing and training, were performed with 8kHz sampled speech files.

5.5 Statistical Significance

In experimental design of speech recognition systems where the performance of one design is compared to the performance of another, there is a need to attach some statistical significance to any inferences that may be drawn about the performance of one design over the other. It is not convincing to conclude that one design is better than another on the basis of WER they produce. This is because such conclusions do not take into account uncertainties that are there in such experimentations. The *matched pairs test* as applied to speech recognition was suggested by Gillick and Cox [73] for testing the statistical significance of any differences in performance between two designs. Of the two presented in the paper [73] (the other one is the McNemar's test), the matched pairs test is the most suitable for continuous speech recognition experiments. This method is best used in experiments whose output can be divided into segments. Errors in one segment are supposed to be independent of errors in another segment. In continuous speech recognition output these segments are utterances in a test set. The following is the description of the matched test method.

Let N_A^i be the number of errors made by design or algorithm A when used to recognize utterance number i in the test set and N_B^i the number of errors made by design B in recognizing utterance i. Let $Z^i = N_A^i - N_B^i$ for $i = 1, 2, 3, \ldots, n$ be the difference between the numbers of errors made by design A and B on utterance i where n is the number of utterances in the test set. Z^i can be assume to have a normal distribution. Let μ_z be the average difference between errors made by algorithms A and B. By implication the estimate of μ_z is,

$$\hat{\mu}_z = \sum_{i=1}^{n} Z^i/n \tag{5.1}$$

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

$$\hat{\sigma}_z^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Z^i - \hat{\mu}_z)^2 \tag{5.2}$$

Then the test statistic is defined as,

$$W = \frac{\hat{\mu_z}}{(\hat{\sigma}_z/\sqrt{n})} \tag{5.3}$$

If n is large enough (> 50), W will approximately have a Normal distribution with a unit variance.

The aim of the test is to prove or disprove the hypothesis that the performances of two compared recognition systems are the same. This hypothesis is equivalent to saying there is no difference in errors produced by design A and those produced by design B i.e. that $Z^i = N_A^i - N_B^i = 0$. This equation implies that $\mu_z = 0$. We therefore define the null Hypothesis as $\mathbf{H_o} : \mu_z = 0$. This means that the distribution of W also has zero mean.

Now to test the $\mathbf{H_o}$ hypothesis we compute $z(1-(\alpha/2))$, where α is a chosen significance level. Typical values for the significance level are 0.1, 0.05, 0.01 or 0.001. $z(1-(\alpha/2))$ means $\int_{-\infty}^{z} g(x)dx = 1-(\alpha/2)$ where $g(x)$ is a Normal distribution function. This is a *one-tailed* test. We do not reject the null hypothesis with a confidence of $(1-\alpha)$ if $W \leq z(1-(\alpha/2))$. We infer that one algorithm or design performs better than the other if $W > z(1-(\alpha/2))$. For the significance level $\alpha = 0.001$ (99.9% confidence) $z(1-(\alpha/2)) = 3.091$, for $\alpha = 0.01$ (99% confidence) $z(1-(\alpha/2)) = 2.327$, for $\alpha = 0.05$ (95% confidence) $z(1-(\alpha/2)) = 1.645$ and for $\alpha = 0.1$ (90% confidence) $z(1-(\alpha/2)) = 1.282$.

The matched pairs test is used in this work to determine the confidence with which any

notable improvement brought about by the investigated filter can be accepted.

5.6 Base-line Results and Discussions

Performance of systems in this work are evaluated in % WER (word error rates) defined as,

$$W = \frac{D + I + S}{N}\% \tag{5.4}$$

where D stands for the number of word deletion errors made by the system, I the number of insertion errors, S the number of substitution errors and N the total number of words in the test set used.

Base-line results, obtained when the base-line system was tested with clean speech and GSM speech with different numbers of mixture densities per HMM state (1 - 9), are depicted in Figure 5.2. The degradation of in performance of systems on GSM speech is clearly depicted in the gap between the clean speech and the GSM speech curves. The best performance of the system on clean speech occurred at 4 mixtures where the WER was 7.2%. At the same number of mixtures the best performance of the system given GSM speech test set is 33.97%. Any improvements brought about by the pre-filtering method under investigation in this work is evaluated against this WER (33.97%).

5.7 Summary

In this chapter the design of the base-line system together with the procedure used in building it have been described. This procedure is followed in building all the other systems built for experimentation in this work. In the next chapter channel normalization techniques and experiments performed with some of them are discussed.

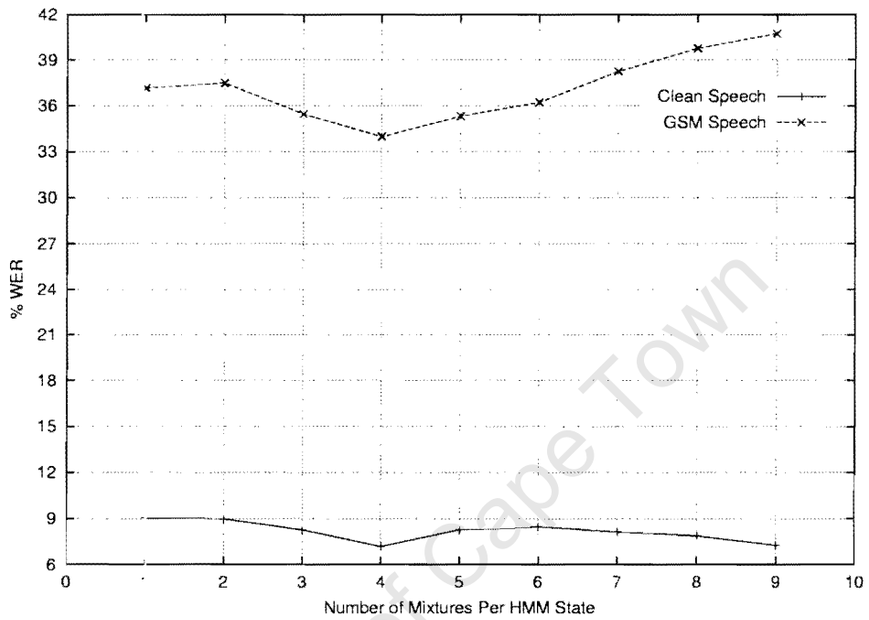


Figure 5.2: Performance of the base-line system on both clean speech and GSM speech.

Chapter 6

Channel Normalization: Theory and Experiments

Introduction

In this chapter, various techniques used for channel compensation are briefly discussed and results of experiments performed with some of them are given (specifically RASTA filtering and CMS). These were tested for their effectiveness in compensating for channel effects in real GSM speech. Results of these tests are listed, discussed and compared. The HTK [63] implementation of CMS was used and the RASTA filter implementation was by Hynek Hermansky [74].

6.1 Channel Normalization

Voice channels tend to exhibit properties of a Linear Time Invariant (LTI) filter acting on a speech signal as it propagates through the channel. This filter-like operation of the channel distorts speech signals transmitted through the channel. There are also 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 convolutional and additive in nature. In the form of an equation a voice channel affected speech signal is often expressed as,

$$y(t) = h(t) * s(t) + n(t) \tag{6.1}$$

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

The aim of channel normalization is to eliminate or compensate for channel effects on speech in order to improve recognition performance of speech affected by a voice channel. There are three points in the recognition process at which channel normalization is normally performed, namely, prior to feature extraction, after feature extraction and during pattern recognition. There are generally three classes of channel normalization techniques, filter based (or pre-processors), feature modification and model modification techniques [75]. Filter based techniques often operate on the signal prior to feature extraction in an attempt to condition a channel speech signal somehow to reduce channel effects on it. 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 so as to reduce channel effects on speech recognition.

Two well-known channel normalization techniques, RelAtive SpecTrAl filtering [17] and Cepstral Mean Subtraction (CMS) [16], were experimented with. Both these methods are feature modification techniques. These methods were chosen due to their popularity and their well-known effectiveness in speech recognition. Each of the two techniques is briefly explained in the following sections.

6.2 Cepstral Mean Subtraction (CMS)

The Fourier spectrum of Equation 6.1 is expressed as,

$$Y(f) = S(f)H(f) + N(f) \tag{6.2}$$

The log of the term $S(f)H(f)$ becomes,

$$\ln(S(f)H(f)) = \ln S(f) + \ln H(f) \tag{6.3}$$

From Equation 6.3 it is noted that the channel transfer function which is convolutional in the time domain is additive in the log spectral domain.

As mentioned in Chapter 2 one of the most popular feature transformations in speech

recognition is the computation of cepstral coefficients. Also mentioned in Chapter 2 is the fact that the cepstrum, from which cepstral coefficients are computed, is computed by taking the log of spectral magnitudes and computing the inverse Fourier transform of the log spectrum (i.e. Equations 6.3 and ??). This means in the cepstral domain features, as in Equation ??, the channel transfer function is additive and can therefore be subtracted from cepstral features. In CMS, the cepstral features relating to the channel transfer function is estimated as the mean vector of cepstral vectors coefficients. To normalize the channel, this estimate (mean) vector is subtracted, hence the name Cepstral Mean Subtraction. For a better estimate of the cepstrum of the transfer function as a mean, a lot of data is required from which to compute it. In this work the estimate is obtained from speaker clusters (i.e. there is one specific cepstral mean vector for all the utterances of each speaker). CMS is very simple to implemented and is well-known to be effective. Virtually all speech recognition applications use one form of it or another. The only one weakness of CMS is that it does not deal with additive noise, it only addresses convolutional noise as may be seen in the above equation-based analysis.

The baseline system of this work implements the speaker cluster CMS (See Section 5.2). In the next section RASTA filtering is discussed.

6.3 RASTA

Speech is produced by movements of the vocal tract at different changing rates. These movements and their rates of change 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 [17]. 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 [17]. RASTA processing was developed as an extension of the PLP feature extraction method.

Frequency characteristics of communication channels are often fixed or slowly varying. If feature extraction methods used for representation of speech can be robust to these frequencies then the effects of the channel will have been normalized successfully. This is the motivation behind RASTA processing. It is based on the principles applied in the case of the PLP feature extraction method. The power spectrum of a given frame of speech is computed from which an auditory spectrum (critical band spectrum) is computed as

in the PLP case [2]. This auditory spectrum is transformed by using some form of a compression function. The log of the resulting transform is put through a RASTA filter. The output of the RASTA filter is decompressed using an inverse of the compression function applied on the auditory spectrum. From the resulting spectrum linear prediction (LP) model coefficients are computed; these are extracted features. From these steps of RASTA processing it is clear that it is the same as the PLP feature extraction except that RASTA processing applies a RASTA filter whereas the the PLP operation doesn't. The transfer function of the RASTA filter used is,

$$H(z) = 0.1z^4 \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.94z^{-1}} \tag{6.4}$$

There are two forms of RASTA processing. They differ from each other with regard to the compression and decompression functions they use. The standard one uses simple logs (log-RASTA) whereas the other one uses Lin-logs (Lin-log-RASTA). The compression function used in Lin-log-RASTA case is defined as,

$$y = \ln(1 + Jx) \tag{6.5}$$

The factor J is a positive constant that depends on the SNR of a given signal. This is the reason Lin-log RASTA processing is also known as J-RASTA processing. The inverse of 6.5 which is meant to be used for decompression is,

$$x = \frac{e^y - 1}{J} \tag{6.6}$$

However, this decompression function can result with negative values. For this reason a very similar decompression function is used which is,

$$x \approx \frac{e^y}{J} \tag{6.7}$$

The J-RASTA is claimed to address both the channel distortion and additive noise. Both the CMS and RASTA techniques were tested on a real GSM speech recognition task. In the next section, results of these tests are mentioned and discussed.

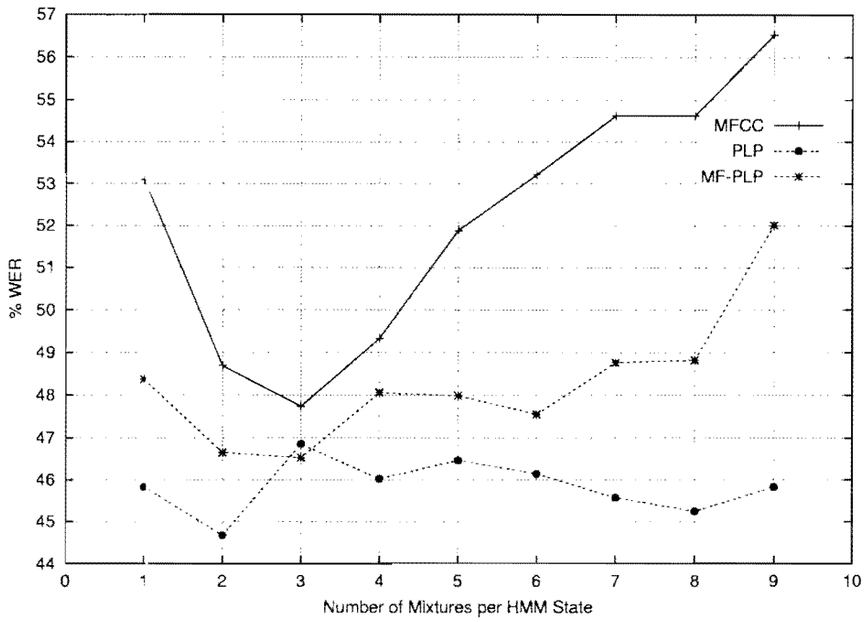


Figure 6.1: Results of systems with MFCC, PLP and MF-PLP feature extraction methods tested without any channel normalization.

6.4 Results and Discussions

CMS was applied on all three tested feature extraction methods, the PLP, MFCC and MF-PLP. RASTA is applied on the PLP only since it was designed and developed to work with the PLP feature extraction method [17]. The front-ends including the MF-PLP and the MFCC are exactly the same as the base-line front-end outlined in Section 5.2 except that they have no channel normalization, in this case CMS. The PLP front-end is also the same as the base-line front-end except that it does not have VTLN and channel normalization and is made up of 8 PLP features plus the energy term.

The results of these tests are divided into three subsections, namely, results without any channel normalization, results with the RASTA channel normalization and results with CMS channel normalization.

6.4.1 No Channel Normalization

Figure 6.1 depicts results obtained from the three feature extraction methods tested without channel normalization. The lowest WER produced by the MFCC is 47.74% at 3 mix-

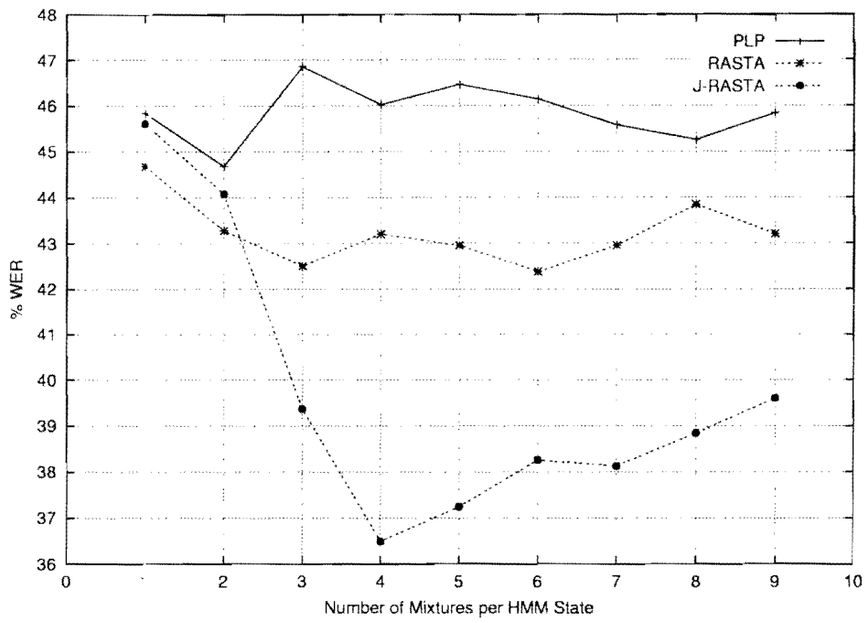


Figure 6.2: Recognition performance for both log-RASTA and J-RASTA channel normalization compared with PLP.

tures, the PLP's is 44.68% at 2 mixtures and the MF-PLP 46.53% at 3 mixtures. These results show that the PLP is more robust to GSM speech than the MFCC and MF-PLP which is in line with results obtained in previous research work [76, 77].

6.4.2 RASTA

In this subsection results of the RASTA channel normalization techniques are presented. For the J-RASTA it was found in the results published in [78] that the value of $J = 10^{-6}$ gives the best results. It is this value that was used to obtain the J-RASTA results.

Figure 6.2 depicts the results of both the J-RASTA and the normal log-RASTA. From these results it is evident that both versions of RASTA filtering improve robustness of recognition systems to GSM speech. log-RASTA produced the lowest WER rate of 42.38% at 6 mixtures. This is 5% decrease in WER compared to the lowest produced by the simple PLP which is 44.68% at 2 mixtures. J-RASTA produced the lowest WER of 36.5% at 4 mixtures. This is 18% reduction in WER compared to the lowest produced by the PLP. These results prove the superiority of J-RASTA over the simple log-RASTA which is in line with the results obtained in [17].

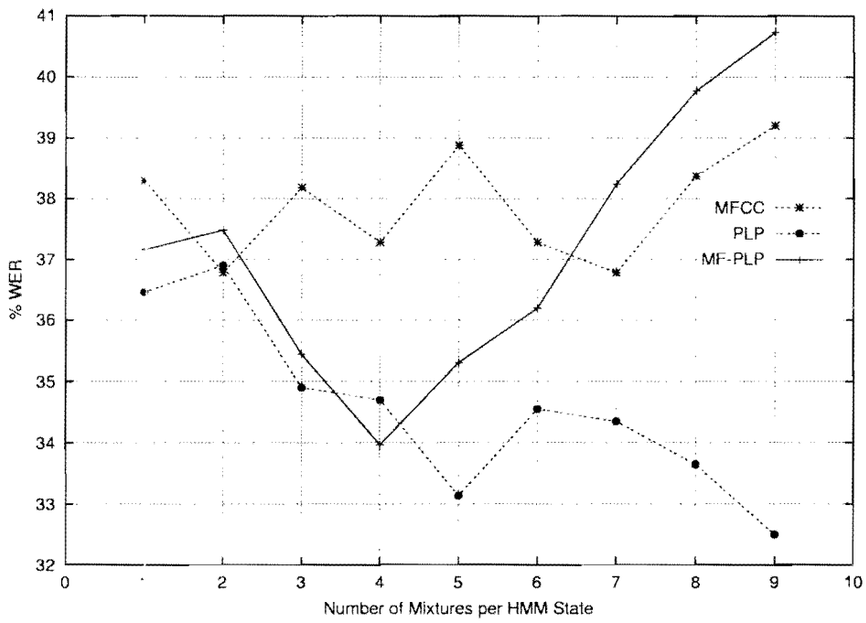


Figure 6.3: Recognition performance with CMS applied on MFCC, PLP and MF-PLP feature extraction methods.

6.4.3 CMS

For these experiments and all others that involve CMS the cepstral mean was calculated per speaker, i.e. the cepstral mean vectors for each speaker utterance were computed from all the utterances spoken by that speaker.

Figure 6.3 shows recognition results obtained when CMS was applied on the MFCC, PLP and MF-PLP feature extraction methods. The MFCC produced the lowest WER rate of 36.78% at 2 mixtures, PLP's lowest is 32.5% at 9 mixtures and MF-PLP's is 33.97% at 4 mixtures.

From the results outlined above the dominance and the effectiveness of CMS is evident. The lowest WER rate produced by RASTA filtering is 36.5% compared to 32.5 of CMS (The two can only be compared on PLP results since RASTA is applied on PLP only). This is an 11% difference. It is this effectiveness over other channel normalization techniques and its simplicity that has causes CMS to be the current channel normalization of choice in speech recognition.

These results also show that the CMS applied on the PLP produced the best overall results. Even though this was also the case in the development of the 1996 HTK Switchboard system

[30, 65], the MF-PLP was found to be more consistent whereas PLP results were somewhat mixed and consequently could not be relied upon. This is the reason the MF-PLP was preferred over normal PLP for that system and other systems [14, 15, 31, 32, 33].

6.5 Summary

In this chapter, concepts of channel normalization have been mentioned and briefly discussed. Theories of CMS and RASTA processing were discussed in detail. Experiments were conducted with the RASTA and CMS channel normalization techniques to determine if they brought about any significant improvements on GSM speech recognition performance and also to compare their performances. Results obtained from these experiments, as outlined in this chapter, proved that the two tested channel normalization techniques do improve GSM speech recognition performance. By comparison CMS proved the better of the two in all tested front-ends. As a byproduct of this comparison, the MF-PLP, PLP and MFCC were compared. Although normal PLP produced better results previous research that led to the development of MF-PLP claim it is inconsistent [30, 65]. These results qualify the chosen base-line front-end. In the next section the pre-filter implemented in combination with CMS intended to further improve the GSM speech recognition performance of the base-line system is discussed and results of experiments conducted with it are mentioned and discussed.

Chapter 7

Variable Filter:Theory, Experiments and Results

Introduction

The variable filter proposed in this work to improve robustness of recognition systems to GSM speech is widely used as a speech enhancement post-filter in low-bit speech coding communications networks. Its success in telecommunications is clearly seen in the fact that different variations of it have been incorporated into several national and international speech coding standards [79]. Among these standards are the U.S. Federal Standard 4.8kbps CELP (FS1016) [80], North American digital radio standard 8kbps VSELP (IS-54) [81], the Japanese digital cellular radio standard 6.7 kbps VSELP (JDC) and the CCITT standard 16kbps low-delay CELP [82, 83]. The filter is used to improve perceptual quality of coded speech by reducing perceived noise introduced by low-bit speech codecs. In this work, it is used for the same purpose on GSM speech.

This chapter begins with a brief motivation for the experiments conducted with the variable filter. The motivation is followed by a discussion of the theory of auditory masking which served as a philosophy behind the development of the filter. This is followed by a detailed discussion of the filter and its different stages. Lastly, experiments conducted with this filter together with the results they yielded are mentioned and discussed.

7.1 Motivation for Experiments

In the GSM standard no post-filtering is used because the perceptual quality of GSM speech is considered good enough without employing a speech enhancement post-filter. However, good perceptual quality does not mean good quality speech for speech recognition systems. It is for this reason that in this work, the variable filter which is used as a post-filter in telecommunications is used as a pre-filter to speech recognition systems. This is an attempt to improve the quality of GSM speech thereby improving the recognition thereof. Its function is to reduce GSM speech coding noise prior to recognition. For these experiments the filter and all its stages were coded from scratch in C programming language.

7.2 Theory of Auditory Masking

Given a sinusoidal wave with a certain frequency and intensity, there exists in a normal human hearing system a masking threshold function such that if noise, whose power spectrum is below that of this masking threshold function, is added to the sinusoid the noise will be inaudible (masked) [84]. A short segment of a speech signal can be regarded as many sinusoid at different frequencies superimposed. For each one of these sinusoids there is a noise masking threshold function in a normal hearing system. These noise masking threshold functions are assumed to be also superimposed to result in a noise masking threshold for the segment of speech as a whole [79]. This implies that any noise with a power spectrum below the power spectrum of the overall noise masking threshold at all frequencies will be completely inaudible to the listener listening to the segment of speech. This overall noise masking threshold follows to an extent spectral peaks and valleys of, not only the envelope of the speech signal segment spectrum, but also the pitch harmonic peaks for voiced speech [79].

Therefore, to reduce speech coding noise a speech coder must push the noise spectrum below the spectrum of the noise masking threshold function spectrum. This must be done at all frequency components over the time span of the speech segment as the speech spectrum changes. If this is achieved, speech coding noise will be completely inaudible to the human ear. However, in practice this is difficult to achieve. Suppressing certain noise frequencies tends to amplify other noise frequency components [54]. In speech perception, formant regions of speech are more important than non-formant regions. To take advantage of this fact, a good approach to speech coding noise reduction is to sacrifice non-formant

regions for formant regions. This can be done by attenuating noise in formant regions so that it is in these regions below the noise masking function threshold. Doing this results in amplification of noise components in non-formant regions which may exceed the masking threshold. However, this noise in these regions can be attenuated using post-filtering techniques. This is the strategy employed by the variable filter the use of which is proposed in this work. In the following Section this variable filter is discussed in detail.

7.3 The Variable Filter

There are two filtering stages in the filter, the short prediction and the long term prediction stages. The short term prediction (STP) attenuates noise in formant regions and in the process amplifies it in inter-formant regions. The long-term predictor (LTP) attenuates noise components in inter-formant regions. The short term and long term prediction filters can be used as two independent filters but they are often used in tandem. In such a set-up, the LTP is used to redress the negative effects of the STP by attenuating noise components in inter-format regions where the STP amplifies them. Due to their independence in operation the two filters are discussed separately in subsections below.

7.3.1 Short Term Predictor

The transfer function of the STP filter is defined as,

$$\begin{aligned} F(z) &= \frac{A(z/\beta)}{A(z/\alpha)} \\ &= \frac{1-\sum_{k=1}^{p} a_k \beta^k z^{-k}}{1-\sum_{k=1}^{p} a_k \alpha^k z^{-k}} \end{aligned} \tag{7.1}$$

where $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$ (an LPC analysis filter), $\alpha \geq 0$ and $\beta \leq 1$. The choice of parameters α and β depends on the speech coding scheme for which the filter is used and the bit rate of that speech codec [85]. In this work the LPC order p was set to 10. The frame size over which the LPC coefficients, a_k, are computed was set to 200. Throughout the rest of this document, the term variable filter refers to the STP filter in the form of Equation 7.1.

When $\beta = 1$ and $\alpha < 1$ then,

$$F(z) = \frac{A(z)}{A(z/\alpha)} \tag{7.2}$$

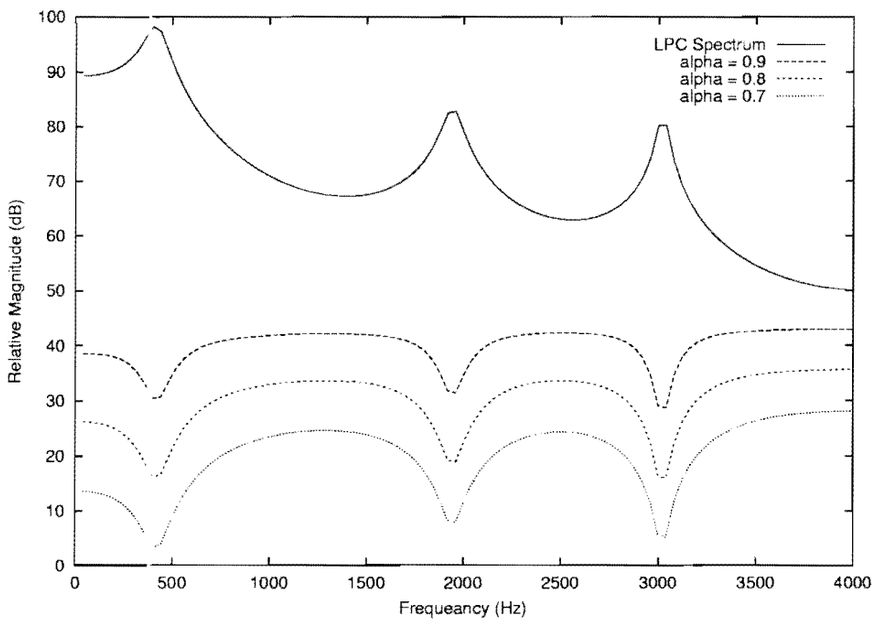


Figure 7.1: Spectra of $1/A(z)$ and $A(z)/A(z/\alpha)$ for $\alpha = 0.9$, $\alpha = 0.8$ and $\alpha = 0.7$.

With such values, the filter is exactly the same as the error weighting filter of Equation 4.24 [54] used in the GSM speech codec and other analysis-by-synthesis speech codecs. Throughout the rest of this document the term error weighting filter refers to the STP filter in the form of Equation 7.2. In the GSM codec the filter is used to shape (weight) the flat noise spectrum in such a way that noise components in formant regions of the speech spectrum are attenuated. The noise in consideration at this point is quantization noise. Even in its application in post-filtering it attenuates the speech coding noise in formant regions with the negative result of amplifying it in inter-formant regions. It is important to note that this attenuation results in attenuation of speech spectral peaks (formants) too which is undesirable but cannot be avoided.

Figure 7.1 shows the LPC spectrum for a voiced segment of speech and the spectra of the the error weighting filter for different values of α. These spectra show the desired attenuation in formant regions and amplification in non-formant regions. Experiments were conducted with the error weighting filter independently to determine the effects of attenuating GSM noise in formant regions on the robustness of recognition systems to GSM speech. Results of these experiments are given in Section 7.4.

Figure 7.2 shows the LPC spectrum for a voiced segment of speech together with the spectra of the post-filter in Equation 7.1 for different values of α and β. For $\alpha = 0.8$ and

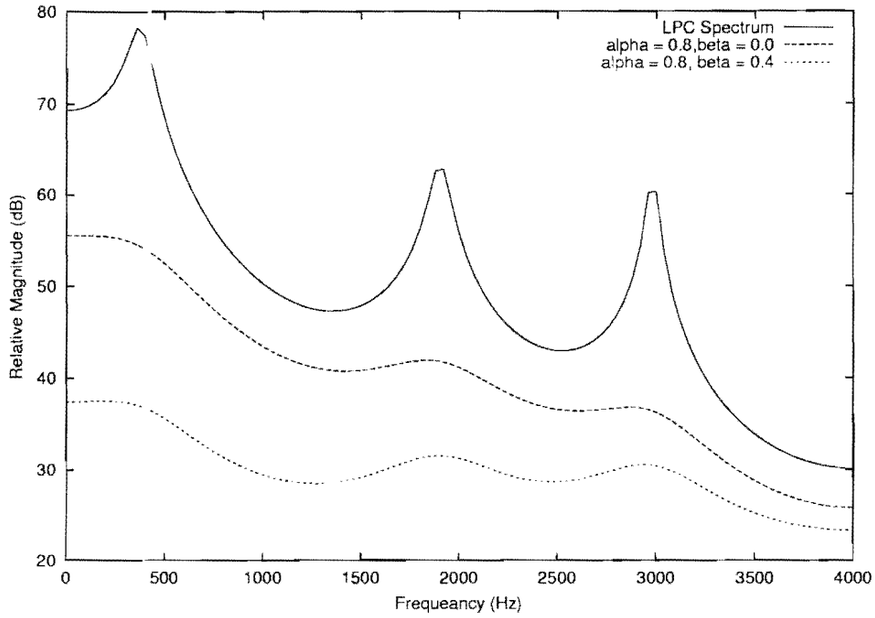


Figure 7.2: Spectra of $1/A(z)$, $1/A(z/0.8)$ and $A(z/0.5)/A(z/0.8)$.

$\beta = 0$ (which is the weighted synthesis filter) the filter shows a low-pass spectral tilt. In low-bit speech coding communications, this is a problem since the weighted synthesis filter applied on coded speech will emphasize lower frequencies and attenuate higher frequencies leading to a poorer perceptual quality even though perceived noise is reduced. To ease this spectral tilt the values of $1 > \beta > 0$ are used. The spectrum of the post-filter for values $\alpha = 0.8$ and $\beta = 0.5$ on Figure 7.2 shows this eased spectral tilt compared to that of the weighted synthesis filter. Experiments with this form of the variable filter have been performed to determine the effects of the eased spectral tilt on the robustness of recognition systems to GSM speech. To further remove the spectral tilt, the filter is often implemented in this form,

$$F(z) = (1 - \mu z^{-1}) \frac{A(z/\beta)}{A(z/\alpha)} \tag{7.3}$$

where $\mu < 1$. In this form, the variable filter is sometimes referred to as the modified variable filter. This term is adopted and used in the rest of this document. Note that the term $(1 - \mu z^{-1})$ is a high-pass filter used as a pre-emphasis filter in many speech recognition systems' front-ends (See Section 2.3). As in the case of pre-emphasis this filter in the modified variable filter emphasizes formants which are at high frequencies. Experiments were also conducted with the modified variable filter; results of these are also

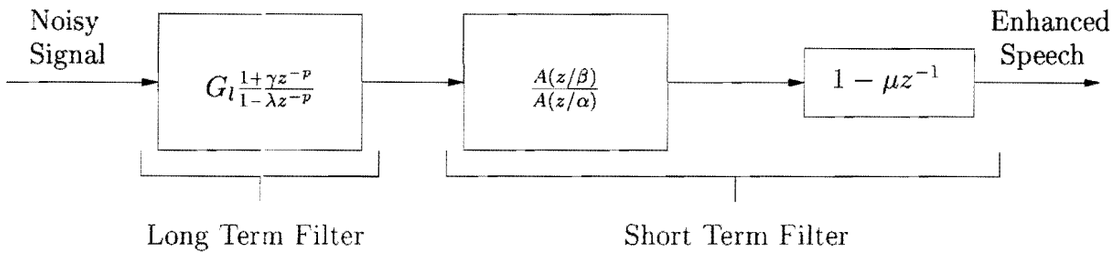


Figure 7.3: Implementation of the variable filter with both the short-term and the long-term filters.

mentioned in Section 7.4.

7.3.2 Long Term Predictor

As mentioned, the long predictor aims to to attenuate noise frequency components between pitch harmonic peaks, which effectively suppresses noise in inter-formant regions [79]. A one-tap long term or pitch predictor defined as $1-gz^{-p}$ has a corresponding pitch synthesis filter defined as $1/(1-gz^{-p})$ where g is the filter gain and p the pitch period in terms of the number of samples. When g is positive, the poles of the pitch synthesis filter are at phase angles $0, 2\pi/p, 4\pi/p \ldots, (p-1)\pi/p$ which correspond to frequencies of the pitch harmonics. To achieve desired spectral peaks at pitch frequencies at these phase angles the pitch synthesis filter can be defined as $1/(1-\lambda z^{-p})$ where λ is a suitably chosen constant. Zeros may also be added to the pitch synthesis filter to gain more flexibility and control of its frequency response. To keep spectral peaks at the correct frequencies the zeros must be placed at phase angles corresponding to valleys between pitch harmonics which are $\pi/p, 3\pi/p, ..., (2p-1)\pi/p$. The polynomial $(1+\gamma z^{-p})$ has zeros at these phase angles. Using these two polynomials the long-term filter is defined as,

$$H_l(z) = G_l \frac{1+\gamma z^{-p}}{1-\lambda z^{-p}} \tag{7.4}$$

where G_l is the filter gain. The choice of variables γ and λ depend on the speech coding scheme for which the filter is used. They are determined from the following equations [79],

$$\gamma = C_z f(x), \quad \lambda = C_p f(x), \quad 0 < C_z, C_p < 1 \tag{7.5}$$

where,

$$f(x) = \begin{cases} 0 & x < U \\ x & U \leq x \leq 1 \\ 1 & x > 1 \end{cases} \tag{7.6}$$

and x is the gain (or sum of the gains) of the LTP filter in the analysis block. It is used in this case as a voiced/unvoiced speech indicator. U is a threshold set to determine which segments of speech are voiced and which are unvoiced. C_z and C_p are factors which control positions of the zeros and the poles of the filter respectively.

The filter gain, G_l, is calculated as,

$$G_l = \frac{1 - \lambda x}{1 + \lambda x} \tag{7.7}$$

The detailed derivation of this gain is given in [79].

Since the advent of this post-filter many variations of it have been suggested and used in many different speech coding schemes. Since there is no prior knowledge of the most suitable implementation of this filter, different variations of it were experimented with. This long term filter was not tested alone, it was tested in tandem with the short term predictor. Figure 7.3 depicts this implementation arrangement of the short-term and long-term predictor filters.

7.4 Results and Discussions

In this section, results of experiments conducted with the variable filter are outlined and discussed. First the results of the short term prediction filtering are presented followed by those of the combination of the short term prediction filtering and long term filtering.

Three forms of the filter were considered each characterized by the values assigned to its variable parameters. These are the error weighting filter suggested by [54] (Equation 7.2, $\beta = 1$ and α varied), the variable filter (Equation 7.1, α fixed and β varied) and the modified variable filter (Equation 7.3, α and β fixed and μ varied). Results of each of these are listed and discussed separately in the sub-sections following. Since the best settings for these parameters depend on the bit rate of the speech signals on which the filter is applied [85], they had to be varied to determine the best combination for GSM speech whose bit

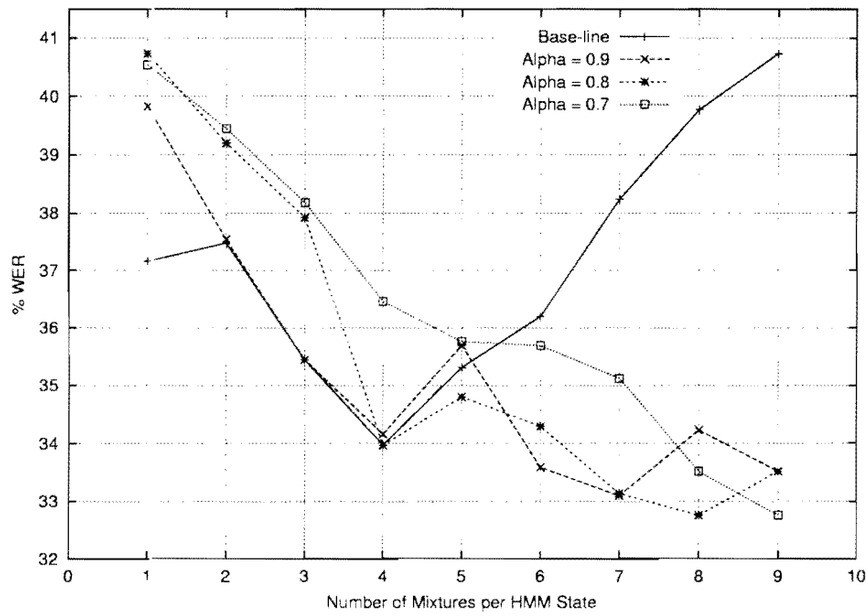


Figure 7.4: Performance of the system with the error weighting filter for different values of α compared to that of the base-line system on GSM speech.

rate is 13kbps.

Different variations of the combination of the STP and LTP filtering were also tried. Variations came in the values assigned to variable parameters of the filters.

7.4.1 Error Weighting Filter

As mentioned in subsection 7.3.1, the error weighting filter is the same as the variable filter with $\beta = 1$ and $\alpha < 1$. In [85], $\alpha = 0.8$ was found to give the best speech enhancement results. This value was chosen as the initial value when the error weighting filter was applied. Other values in the vicinity of this value were also tested to see if they would bring about better performance. The value for α that was found to produce the best recognition results for GSM speech was 0.8.

Performance results of applying the error weighting filter as a pre-filter are shown on Figure 7.4. It be seen on the figure that the error weighting filter brought about an improvement in the recognition rate of GSM speech compared to that of the base-line system. At 4 mixtures the error weighting filter system matched the base-line system with a %WER of 33.97. Beyond 4 mixtures the error weighting filter system proved the better of the two

systems.

At mixture numbers 7,8 and 9 the error weighting filter system produced word error rates lower that the best produced by the base-line system. The best % WER produced by this system was 32.76 at 8 mixtures. This spells out a 4% drop in the error rate produced by the base-line system. Using the matched pairs test this gave 76.48% percent confidence that the improvement was brought about by the error weighting filter. These improvements are attributed to the noise masking ability of this filter for which it was considered in the first place.

One undesirable factor about these results is that the error weighting filter produces peak performance which is higher than that of the base-line system at 8 mixtures. The more mixtures there are per HMM state the more computation power is required during recognition. The discussion of the operation of the filter in Section 7.3 indicates that the filter achieves reduction of noise levels by attenuation of formant regions of a speech signal. Though this is undesirable it cannot be avoided, yet it has to be done moderately. LPC based features, such as the MF-PLP as used in this case, are computed based on formant information contained in speech signals. In cases where formants are not clearly detectable these features tend are less accurate. In an HMM speech recognition system more mixtures will then be required in order to detect the formants. Since, the error weighting filter attenuates formants it makes more HMM mixtures necessary for a higher recognition accuracy. This explains why the accuracy of the tested system is much better given filtered speech than unfiltered at high mixture numbers.

7.4.2 Variable Filter

Results of applying the variable filter in its complete form, as depicted in Figure 7.5, showed that the β value (with α fixed at 0.8) that produces the best results is 0.4. At this value the system shows a steady decrease in the % WER with the increase in the number of mixtures. The system's performance is lower than that of the base-line system at 4 mixtures where the base-line system produced the best performance (i.e 33.97%). Beyond 4 mixtures the system produced error rates even lower than that of the base-line system. The best of these is 31.93% at 7 mixtures. This is a 6% reduction of the WER produced by the base-line system. Using the matched pairs test this gave a confidence of 87.1% in accepting that the filter has improved the performance of the tested system given GSM test data.

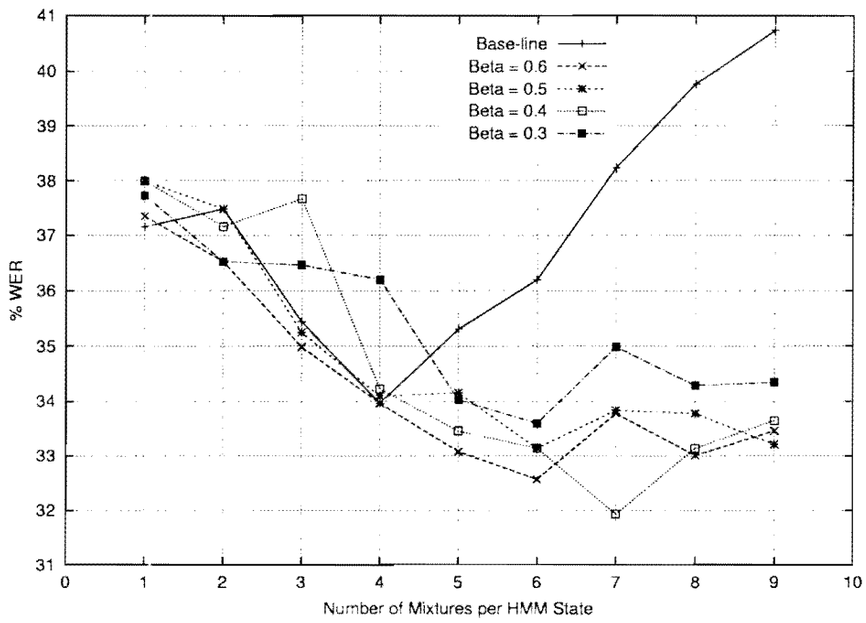


Figure 7.5: Variable filter results for $\alpha = 0.8$ and different values of β compared to the base-line performance.

These results also show that the application of this filter brings about further improvements to those brought about by the error weighting filter. For all mixtures its error rate is below that of the error weighting filter except at 4 mixtures. This improvement over the error weighting filter system is attributed to the fact that the spectral shape of the error weighting filter is tilted to the right so that it emphasizes low frequencies more than high frequencies, acting like a low-pass filter. Values of β less than 1 on the filter ease the tilt, bringing about further enhancement in speech quality and hence higher robustness of the clean speech recognition system to GSM speech.

Just as it is the case with the error weighting filter, more mixtures are required to produce peak performance. This also is due to attenuation of formants that is imposed by the filter.

7.4.3 Modified Variable Filter

The modified variable filter was tested with different values of μ. As shown in Figure 7.6, at all tested values of μ, the filter showed no further improvement to the overall performance of the variable filter. The best performance achieved with this filter is with $\mu = 0.5$. At this value the filter produced the lowest %WER of 33.27% at 9 mixtures.

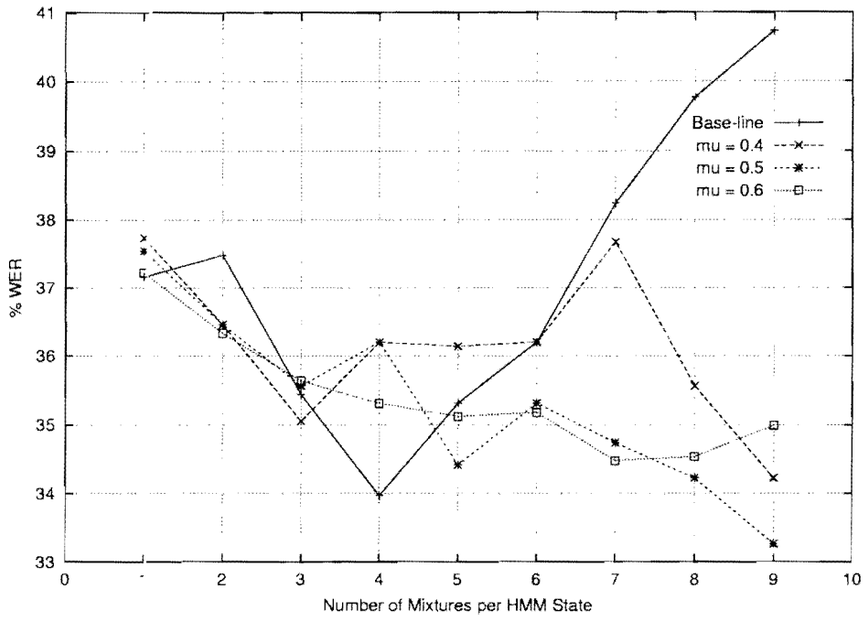


Figure 7.6: Modified variable filter results for $\alpha = 0.8$, $\beta = 0.4$ and different values of μ compared to the base-line system performance.

The modified filter is employed to ease the spectral tilt of the variable filter and this it achieves by emphasizing higher frequencies. This means formants in high frequencies are emphasized. However, noise components at high frequencies are also emphasized. It is this emphasis of high frequency noise components of speech that lead to filter failing to produce a WER lower than that of the variable filter. However, these results could be improved if the noise in inter-formant regions could be suppressed.

Figure 7.7 shows the best results produced by each of all the three forms of the STP prediction filter (error weighting, variable and modified variable filters) compared with one another and against the base-line results. It is clear from the figure that the variable filter with $\alpha = 0.8$ and $\beta = 0.4$ produced the lowest WER. It can also be seen on the figure that the modified variable filter produces rates that are lower than those of the variable filter and of the error weighting filter at mixtures 1 to 3. This is due to the emphasis that the modified filter applies on formants at high frequencies. This emphasis causes formants to be detected by recognition systems even with few mixture densities. This means that if noise components in inter-formant regions and in high frequencies can be suppressed when the modified variable filter is applied then high recognition performance can be achieved at fewer mixtures.

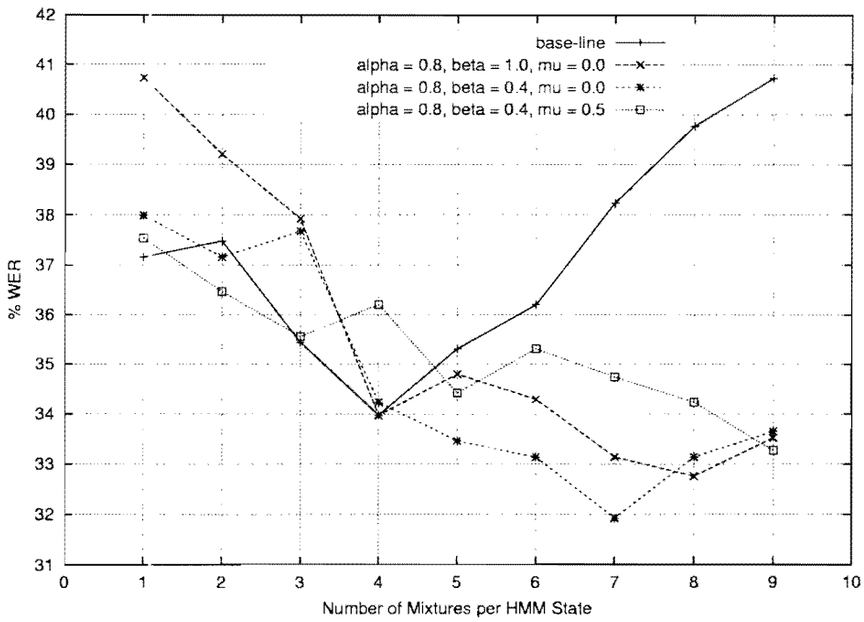


Figure 7.7: Best results obtained from each filter compared with base-line results.

In the following subsection, results of the combination of the STP and LTP filters are presented and discussed.

7.4.4 Combination of the STP and LTP Filters

In the STP filter experiments, it was noted that the STP predictor does improve the robustness of the base-line recognition system to GSM speech especially at high numbers of mixture densities. The LTP filter was included in tandem with the variable STP filter to see if it would bring any further improvements. Results of these tests are presented in this subsection. In all the LTP experiments, $U = 0.6$, $C_z = 0.4$ and $C_p = 0.1$ settings were used. These values are recommended in [79].

Figure 7.8 shows initial results of the implementation of the LTP filter in tandem with the STP filter compared with the best STP filter settings performance and the base-line performance. These results show that the LTP produced better results than the STP filter alone at all mixtures but at 7 mixtures. The lowest WER produced by this combination was 32.57% at 8 mixtures. Improvements shown by this combination, especially at low numbers of mixtures, are attributable to the LTP reducing noise in inter-formant regions so that even though the STP filter attenuates formants they may be detectable even with

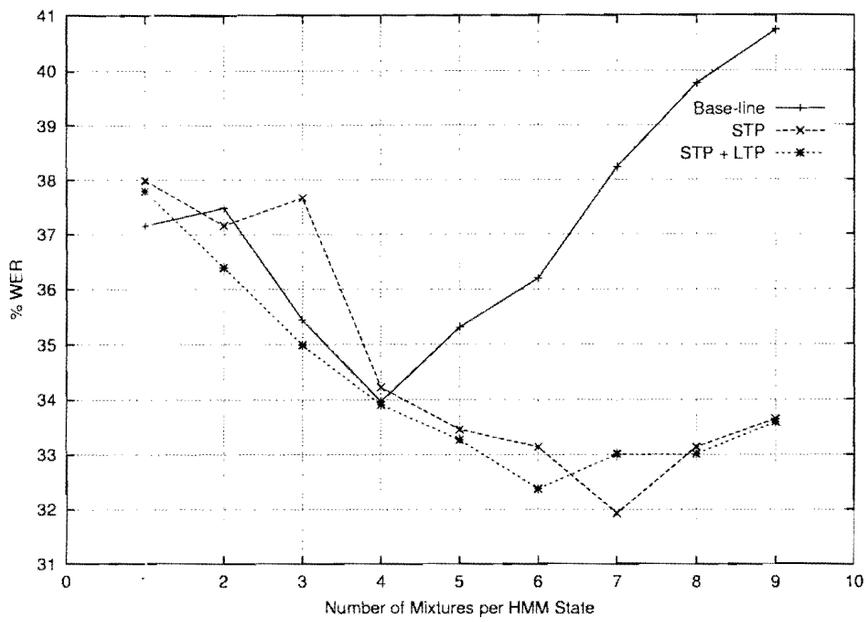


Figure 7.8: Performances of the best STP filter settings with and without the LTP filter compared to the base-line results.

few mixtures.

Having seen these improvements the LTP filter was then used in tandem with the modified variable STP filter. As was mentioned in the discussion of the results of the modified variable LTP filter, this filter emphasizes formants in high frequencies. The LTP filter was put in tandem with the modified variable filter in expectation that it would improve the modified filter's results mentioned in subsection 7.4.3.

Results of this combination are shown in Figure 7.9 and as was expected the LTP filter improved the performance of the system when the modified variable STP filter is applied. This combination produced the lowest WER of 31.80% at 5 mixtures. This WER rate is lower than that of the variable STP filter at fewer mixtures. It is 6% lower than the base-line system's best performance on GSM speech. Using the matched pairs test this gave a 91.1% confidence that the filter has improved performance of the tested system given GSM data.

This reduction in WER at low numbers of mixtures is attributed to the combination of the modified variable filter's emphasis on high frequency formants and the LTP's attenuation of noise components in inter-formant regions. Figure 7.10 shows the LPC spectra for a voiced phoneme /iy/ for clean speech compared to those of the original GSM speech and

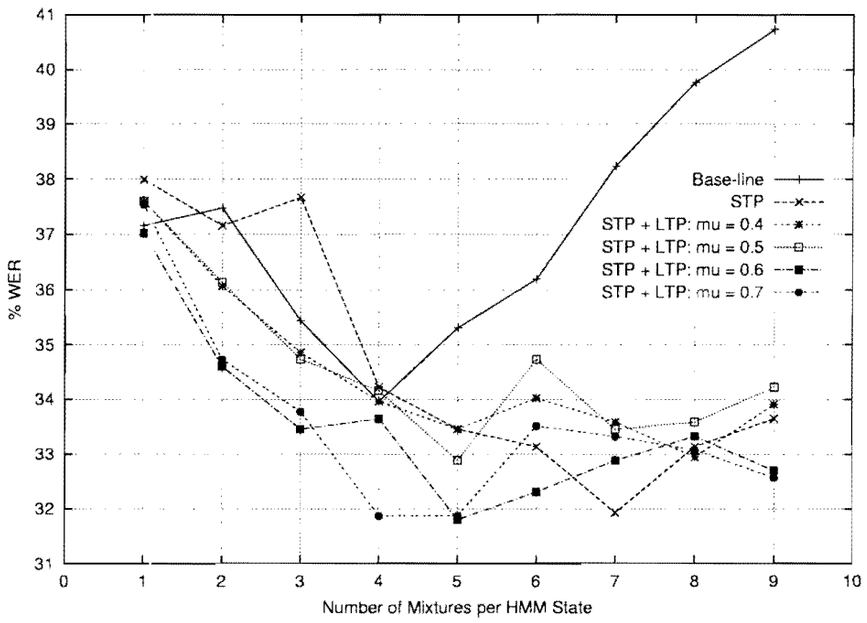


Figure 7.9: Results of the combination of the modified variable STP filter and LTP filter for different values of μ compared with the STP best performance and the baseline performance.

the modified variable STP filter with $\mu = 0.4$ and $\mu = 0.6$ in combination with the LTP filter applied on the GSM speech. From this plot the attenuation of formants caused by the variable filter is evident when the spectra of the original GSM speech is compared to those of the modified variable STP filter combined with the LTP filter. Also, the attenuation of formants due to the GSM channel and GSM speech coding are evidenced by the comparison LPC spectrum of the clean signal to that of the GSM signal. The emphasis on high frequency formants and the attenuation of noise in inter-formant regions (especially between 500Hz and 2000Hz) are also evident on LTP+STP spectra. It is this combination of emphasis of formants and suppression of noise in inter-formant regions that gives good results when the modified variable STP filter is combined with the LTP filter.

In [79] it is suggested that the tested variable filter can be used as an independent speech enhancement filter. An experiment was conducted to test the pre-filter in this capacity. It was applied on an MF-PLP recognition system without any channel normalization. The value of μ used for this filter 0.6. Results of this experiment are shown on Figure 7.11. The lowest WER rate produced by the system using the pre-filter is 44.68% at 3 mixtures compared to 46.53% produced by the system without pre-filtering. This is a 4% reduction

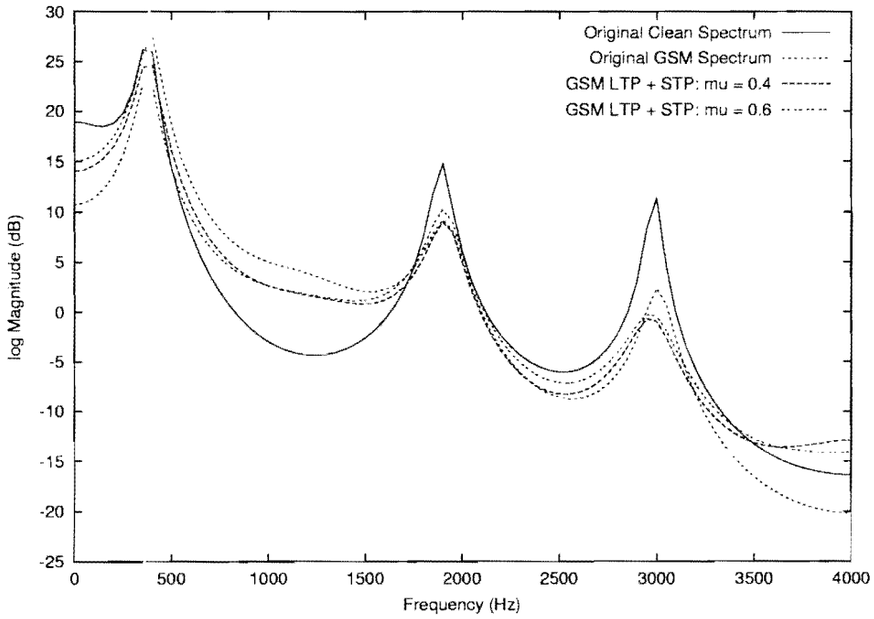


Figure 7.10: Spectra of the LTP-STP combination for different values of μ compared to that of the STP filter, the original clean and GSM speech. These are the spectra of the voiced phone /iy/.

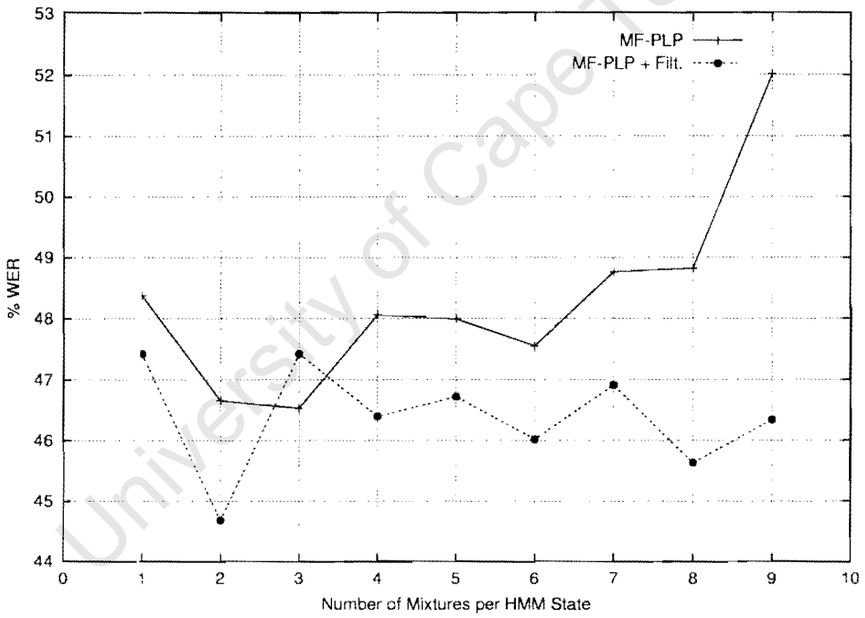


Figure 7.11: Performance of an MF-PLP system without channel normalization compared to its performance using variable pre-filter.

in WER brought by the pre-filter.

7.5 Summary

In this chapter, the theory of auditory masking has been briefly discussed. The definitions of the two components of the pre-filter, the STP and LTP, were given and theory behind them was also discussed. The results of the experiments conducted with different forms of the STP filter alone and of the STP filter combined with the LTP filter were mentioned and discussed. In the next chapter conclusions drawn from these results and suggested further work are given.

Chapter 8

Conclusions

8.1 Summary of work done

The ultimate goal of this work was to improve robustness of speech recognition systems to GSM speech. The two sources of poor GSM speech recognition performance are channel noise and GSM speech coding. Different channel normalization techniques have been devised to address channel effects and improve robustness of speech recognition systems to voice channel speech In this work, CMS and RASTA channel normalization techniques were tested to determine and compare their effectiveness with GSM speech. CMS was found to be the better of the two. A speech enhancement post-filter used extensively in low-bit telecommunications was proposed for use as a speech recognition system pre-filter whose aim is to reduce effects of GSM speech coding on GSM speech recognition performance. The expectation was that this filter combined with channel normalization techniques would bring higher GSM speech recognition rates than when channel normalization is used alone. This pre-filter was tested in combination with the CMS channel normalization in a front-end that was used in some of the CU-HTK Switchboard systems [14, 15]. Conclusions drawn based on the results of this work are discussed in the following section.

8.2 Conclusions

From the results obtained in the experiments conducted in this work, the following conclusions are drawn:

- The use of the post-filter does improve robustness of recognition systems to GSM

speech. The improvement brought about by the use of this filter as a speech recognition system pre-filter is about 6% reduction in WER. This answers the first of the two questions mentioned in the objectives statement of this work.

- Improvements in perceptual quality of speech (for which the filter investigated is mainly used) does necessarily mean improved recognition of the enhanced speech. This conclusion is backed up by the fact that the filter used as a pre-filter has been used extensively in low-bit speech communication networks to enhance speech perceptual quality. In this work, the filter has been used in a speech recognition application and it has brought about improvements in robustness to GSM speech. This answer the second question posed in the objectives statement of this work in the introduction.

- The filter can be used as a speech recognition pre-filter in its own right. Though the filter was used in this work in combination with channel normalization it can be used as a separate independent speech enhancement pre-filter for speech recognition systems. This conclusion is drawn from the result of the experiment conducted with the filter used as a pre-filter for an MF-PLP recognition system where there was no channel normalization. The filter improved the performance of the system by 4%.

In the following section, suggested further work based on this work is given.

8.3 Suggested Further Work

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

- The filter in this work was tested with one feature extraction method (MF-PLP). Since different feature extraction methods compute different features of speech they may be affected differently by the pre-filter. For an example, LPC based feature extraction methods are critically dependent on formants of speech for good performance; how will the formant attenuating effect of this filter affect them?

- Parameters of this filter vary from speech codec to speech codec. Different standards including this filter employ different values. The values used in this work, especially those of the LTP filter are generic ones which have been suggested in literature. These parameters may need to be optimized for GSM speech recognition.

- The filter's performance with parameters (e.g. the LTP synthesis filter gain) directly from the GSM channel codec parameters must be evaluated. In this work these parameters were computed from decoded signals but the performance of the filter may be improved even further if these parameters are obtained directly from the GSM codec parameters.

Bibliography

[1] J. Picone, "Signal modeling techniques in speech recognition," in *Proceedings of the IEEE*, vol. 81, no. 9, September 1993.

[2] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *Journ. Acoustic Soc. America*, vol. 87, no. 4, pp. 1738–1752, 1990.

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

[4] S. Sandhu, "A comparative study of Mel cepstra and EIH for phone classification under adverse conditions," Master's thesis, MIT, 1994.

[5] 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.

[6] S. Euler and J. Zinke, "The influence of speech coding algorithms on automatic speech recognition," in *Proc. ICASSP, Vol 1*, 1994, pp. 621–624.

[7] H. J.M. and S. R.M., "Speech recognition from GSM codec parameters," in *ICSLP-98*, 1998.

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

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

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

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

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

[13] C. Mokbel, L. Mauuary, L. Karray, D. Jouvet, J. Monne, J. Simonin, and K. Bartkova, "Towards improving ASR robustness for PSN and GSM telephone applications," *Speech Communication*, vol. 23, pp. 141–159, October 1997.

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

[15] ——, "CU-HTK April 2002 switch board system," in *Rich Transcription Workshop 2002*, May 2002.

[16] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 254–272, April 1981.

[17] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 578–589, October 1994.

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

[19] S. Young, "The HTK book," July 2000.

[20] R. Salami, L. Hanzo, R. Steele, K. Wong, and I. Wassell, *Mobile Radio Communications.* Pentech Press, London, 1992, ch. Speech Coding.

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

[22] B. Atal and S. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *Journal of Acoustical Society of America*, vol. 50, no. 2, pp. 637–655, March 1971.

[23] N. Levinson, "The Wiener rms error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, pp. 261–278, January 1947.

[24] J. Durbin, "The fitting of time-series models," *Rev. Inst. Int. Stat.*, vol. 28, pp. 233–244, 1960.

[25] R. Steele, *Mobile Radio Communications.* Pentech Press, London, 1992.

[26] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, New Jersey,USA: Prentice-Hall, 1978.

[27] B. Atal, "Linear prediction for speaker identification," *Journal of Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1311, June 1974.

[28] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Comp. Speech and Language*, vol. 1, pp. 109–130, 1986.

[29] H. Hermansky, B. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," in *Proc.IEEE Internat.Conf.Acoust.Speech Signal Process.*, vol. 1, Tampa, 1985, pp. 509–512.

[30] P. Woodland, M. Gales, D. Pye, and 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.

[31] B. Peskin, "Improvement of conversational telephone speech," in *Proceedings of ICASSP'99*, March 1999.

[32] S. Johnson, "Speaker tracking," Master's thesis, MPhil Thesis, Department of Engineering, Cambridge University, UK, 1997. [Online]. Available: citeseer.nj.nec.com/johnson97speaker.html

[33] J. Allan, J. P. Callan, M. Sanderson, J. Xu, and S. Wegmann, "INQUERY and TREC-7," in *Text REtrieval Conference*, 1998, pp. 148–163. [Online]. Available: citeseer.nj.nec.com/48134.html

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

[35] J. Wilpon, C. Lee, and L. Rabiner, "Application of hidden Markov models for recognition of a limited set of words in unconstrained speech," in *Proc. of ICASSP'89*, Glascow, Scottland, 1989, pp. 254–257.

[36] R. Hamming, *Digital Filters (2nd Edition).* Englewood Cliffs, New Jersey,USA: Prentice-Hall, 1989.

[37] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimisation for spoken word recognition," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. ASSP-26, pp. 43–49, February 1978.

[38] C. Myers and L. Rabiner, "A comparative study of several dynamic time-warping algorithms for connected word recognition." *The Bell System Technical Journal*, vol. 60, no. 7, pp. 1389–1409, September 1981.

[39] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," in *Proc. of the IEEE*, vol 73, no. 11, 1985, pp. 1551–1588.

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

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

[42] L. Baum and 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.

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

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

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

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

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

[48] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," in *IEEE Trans. Information Theory, IT-13*, April 1967, pp. 260 - 269.

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

[50] S. Young, N. Russell, and J. Thornton, "Token passing: A simple conceptual model for connected speech recognition systems," *Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept.*, 1989.

[51] P. Kroon, E. Deprettere, and R. Sluyter, "Regular-pulse excitation - a novel approach to efficient multipulse coding of speech," *IEEE Trans. on ASSP*, vol. 34, no. 5, pp. 1054–1063, October 1986.

[52] E. Deprettere and P. Kroon, "Regular excitation reduction for effective and efficient lp-coding of speech," in *Proc. ICASSP'85*, Tampa, Florida, USA, March 1985, pp. 965–968.

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

[54] B. Atal and M. Schroeder, "Predictive coding of speech and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 247–254, June 1979.

[55] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High quality speech at low bit rates," in *Proc. ICASSP'85*, Tampa, Florida, USA, March 1985, pp. 937–940.

[56] B. Atal and J. Remde, "A new model of lpc excitation for producing natural-sounding speech at low bit rates," in *Proc. ICASSP'82*, 1982, pp. 614–617.

[57] T. Aresaki, K. Ozawa, S. Ono, and K. Ochiai, "Multi-pulse excited speech coder based on maximum cross correlation search algorithm," in *Globecom 83*, 1983, pp. 794–798.

[58] M. Berouti, H. Garten, and P. K. P. Mermelstein, "Efficient computation and coding of the multipulse excitation for lpc," in *Proc. ICASSP'84*, 1984, pp. 10.1.1–10.1.4.

[59] B. Konsynski. (2001) History and timeline of GSM. [Online]. Available: http://www.emory.edu/BUSINESS/et/P98/gsm/history.htm

[60] L. Hanzo and J Stefanov, *Mobile Radio Communications.* Pentech Press, London, 1992, ch. The Pan-European Digital Cellular Mobile Radio System-known as GSM.

[61] C. Online. (2003, August) Latest global, handset, base station and regional cellular statistics. [Online]. Available: http://www.celullar.co.za

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

[63] S. Young and P. Woodland. (2002, December) The Hidden Markov Model Toolkit (HTK) version 3.2. [Online]. Available: http://htk.eng.cam.ac.uk/

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

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

[66] DARPA, "The DARPA TIMIT acoustic-phonetic continuous speech corpus, training and test data," 1990.

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

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

[69] 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.

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

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

[72] 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.

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

[74] H. Hermansky. (1991). [Online]. Available: http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/speech/systems/rasta/

[75] R. Bates, "Reducing the effects of linear channel distortion on continuous speech recognition," Master's thesis, Boston University, College of Engineering, 1996.

[76] P. Woodland, M. Gales, and D. Pye, "Improving environmental robustness in large vocabulary speech recognition," in *Proc. ICASSP'96*, Atlanta, GA, 1996, pp. 65–68. [Online]. Available: citeseer.nj.nec.com/woodland96improving.html

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

[78] N. Mahlanyane and D. J. Mashao, "Channel normalization for GSM speech recognition," in *Proc. SATNAC'03*, 2003.

[79] J. Chen and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp. 59–71, January 1995.

[80] J. Campbell, V. Welch, and T. E. Tremain, "An expandable error protected 4800 bps CELP coder (u.s. federal standard 4800 bps voice coder)," in *Proc. ICASSP'89*, May 1989, pp. 735–738.

[81] I. Gersho and M. A. Jasiuk, "Vector sum exited linear prediction (VSELP) speech coding at 8 kbps," in *Proc. ICASSP'90*, April 1990, pp. 461–464.

[82] J. Chen, "A robust low-delay CELP speech coder at 16kbit/s," in *Proc. IEEE Global Commun. Conf.*, November 1989, pp. 1237–1241.

[83] ——, "A low-delay CELP coder for the CCITT 16kb/s speech coding standard," *IEEE J. Selected Areas Commun.*, pp. 830–849, June 1992.

[84] D. O'Shaughnessy, *Speech Communication: Human and Machine.* Reading, MA: Adison-Wesley, 1987.

[85] J. Chen and A. Gersho, "Real-time vector apc speech coding at 4800 bps with adaptive postfiltering," in *Proc. ICASSP'87*, 1987, pp. 2185–2188.