Winter 2006

Automatic Speech Recognition Using LP-DCTC/DCS Analysis Followed by Morphological Filtering

Penny Hix
Old Dominion University

AUTOMATIC SPEECH RECOGNITON USING LP-DCTC/DCS ANALYSIS FOLLOWED BY MORPHOLOGICAL FILTERING

by

Penny Hix

B.S. Electrical Engineering, The University of Texas at Austin

M.S. Applied Mathematics, Old Dominion University

A Dissertation Submitted to the Faculty of

Old Dominion University in Partial Fulfillment of the

Requirement for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
December 2006

Approved by:

Stephen A. Zahorian (Director)

Charlie H. Cooke (Member)

Oscar R. Gonzalez (Member)

David Streight (Member)

ABSTRACT

AUTOMATIC SPEECH RECOGNITON USING LP-DCTC/DCS ANALYSIS FOLLOWED BY MORPHOLOGICAL FILTERING

Penny Hix

Old Dominion University, 2006

Director: Dr. Stephen A. Zahorian

Front-end feature extraction techniques have long been a critical component in Automatic Speech Recognition (ASR). Nonlinear filtering techniques are becoming increasingly important in this application, and are often better than linear filters at removing noise without distorting speech features. However, design and analysis of nonlinear filters are more difficult than for linear filters. Mathematical morphology, which creates filters based on shape and size characteristics, is a design structure for nonlinear filters. These filters are limited to minimum and maximum operations that introduce a deterministic bias into filtered signals.

This work develops filtering structures based on a mathematical morphology that utilizes the bias while emphasizing spectral peaks. The combination of peak emphasis via LP analysis with morphological filtering results in more noise robust speech recognition rates.

To help understand the behavior of these pre-processing techniques the deterministic and statistical properties of the morphological filters are compared to the properties of feature extraction techniques that do not employ such

algorithms. The robust behavior of these algorithms for automatic speech recognition in the presence of rapidly fluctuating speech signals with additive and convolutional noise is illustrated. Examples of these nonlinear feature extraction techniques are given using the Aurora 2.0 and Aurora 3.0 databases. Features are computed using LP analysis alone to emphasize peaks, morphological filtering alone, or a combination of the two approaches. Although absolute best results are normally obtained using a combination of the two methods, morphological filtering alone is nearly as effective and much more computationally efficient.

To my husband Steve Hix and Tom Wolters my friend, colleague, and technical advisor

ACKNOWLEDGEMENTS

First I would like to thank my dissertation adviser, Dr. Stephen A. Zahorian for invaluable patience and advice throughout the research work. His expertise in the area of speech processing allowed me to expand my knowledge during the course of my working in the ECE Speech Communications Lab.

I would also like to thank Dr. Oscar Gonzalez, Dr. Charlie Cooke, and Dr. David Streight for serving on my dissertation committee and for their time and assistance, with a special thank you to Dr. Gonzalez for making sure I had the equipment I needed during the final weeks of my research work.

I would like to thank everyone in the Speech Communications Lab for establishing a nice working environment, with special thanks to Wei Wang for maintaining the fine computer systems in the lab. Even after his graduation, his continued assistance enabled my experiments to run smoothly.

My deepest thanks go to my husband, Steve, for his hard work, encouragement and perpetual patience, to Thomas Wolters whose technical insight, advice, and support have been invaluable, and to my sister Patricia Powers who provided the art work in Figure 3.

Finally, thank you to Dr. Jon Jonsson of NASA Langley Research Center for supporting my research, and for his professional guidance.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 OVERVIEW

Machine signal processing that allows a computer to detect and identify specific words spoken into a microphone or telephone is normally referred to as Automatic Speech Recognition (ASR). Speech recognition systems frequently use recorded speech for training (also referred to as learning). Recorded speech samples are decoded via a combination of signal processing and pattern recognition techniques.

Historically, the primary goal of ASR research has been to obtain 100% accuracy in real-time without constraints. The ideal ASR might be thought of as a speaker independent ASR system with unlimited vocabulary size which is noise robust and that can adapt to changing speaker characteristics. For example, it is quite normal for a speaker to raise the pitch of their voice when they are speaking under stressful conditions. This is one example of what is referred to as the Lombard effect. The ASR systems of today have yet to achieve this ideal. After more than four decades of ASR research, the improved accuracy in quiet or clean speech environments has resulted in only modestly improved performance of ASR in noisy environments. Unfortunately, improvements have not been significant enough to yield recognition accuracy which competes with human performance. The demand for improved ASR systems has driven research to focus on improving Automatic Speech Recognition systems that are noise robust. In quiet environments people are already using ASR dictation systems, and the

United States Government is currently implementing ASR systems to improve and increase the speed of language acquisition for military and government personnel.

Most ASR systems experience performance degradation when attempting to recognize speech in changing environments. Background noise such as room chatter, road noise, or channel noise frequently causes an ASR system to fail dramatically. When recognition performance in noisy environments is improved to above 98%, ASR systems will likely find common applications in command and control systems as well as in many embedded applications. Although the performance of ASR systems in noise has improved, this improvement is most significant only when the recognition task is constrained in some way. Performance is directly dependent on the type of constraint. For instance, accuracy greater than 97% has been achieved on recognition of continuous digits over a microphone channel with no background noise. Performance is even higher for isolated word recognition tasks. In these systems the word to be recognized is surrounded by silence. Therefore, word boundaries are clearly defined and coarticulation effects, which degrade the performance of continuous digit recognition systems, do not contribute to decreased accuracy. Continuous and isolated word recognition tasks are examples of small vocabulary speech recognition systems, and are currently being implemented in clean and noisy environments.

One technique for isolated word recognition is to attempt to recognize whole words. Phonemes are the basic speech unit, but they can be difficult to

recognize. Therefore, some isolated word recognition systems treat each whole word as a basic speech unit. When an ASR system attempts to recognize continuous speech coarticulation effects blur the word boundaries, thus increasing the difficulty level. The Lombard effect adds additional challenges for an ASR system. Speaker accents also introduce variations to which ASR systems have difficulty adapting. These speaker variations combined with environmental differences further increase the challenge of the recognition task.

Speaker dependent systems are trained to recognize voice characteristics of individual speakers. Current speaker-dependent systems normally require approximately 15 minutes of training speech from each speaker; the speaker-specific speech is in addition to the potentially several hours of training speech used to develop the baseline recognizer. During the speaker-specific "enrollment" period, the speaker is placed in a quiet environment and is directed to speak a specific set of words or sentences. The recorded speech samples are then used to train the system to adapt to the voice characteristics of the specific speaker. With this type of system, larger vocabularies are possible but performance is normally degraded to a range of 90% to 95% word accuracy. On the other hand, as an indication of progress that has been made in ASR research, speaker-independent systems such as Large-Vocabulary Continuous Speech Recognition (LVCSR) systems perform quite well in a laboratory. They use a large vocabulary and are trained on a large set of speakers. As a result, individual speakers are not required to provide speech samples for training the recognizer. System complexity and computational demand are directly dictated

by the size of the vocabulary, as is the recognition accuracy. Even with improvements, accuracy in real environments is frequently no greater than 87% [1], and is further degraded in noisy environments. An additional challenge for LVCSR systems is processing time, which can be hundreds of times greater than real-time.

Clearly current Automatic Speech Recognition still presents many challenges. Noisy speech is one of the many unresolved problem areas for Automatic Speech Recognition systems. Even if speech processing is limited to focusing only on the noise in speech, it is impossible to consider all combinations of types and levels. This continuum causes major performance issues for ASR systems. One of the critical problems in this area occurs when acoustic information learned through training is mismatched with the test set. The result is seriously degraded ASR performance. Environmental acoustic mismatch is caused by differences in background noise. On the other hand, channel mismatches occur when transmission conditions vary. The speaker himself can be the source of other types of mismatch depending upon the stress conditions imposed on him or her. As mentioned previously, speakers have a tendency to speak louder, or raise their pitch, in order to compensate for local conditions. Numerous researchers have attempted to build systems that compensate for speech variability [2, 3]. A significant portion of research has also focused on adapting models to noisy speech before the recognition phase; this approach attempts to address variability in the environment. Limited data for adaptation, noise that is highly non-stationary, and insufficient computing power for

adaptation are some of the limitations that must be addressed when recognizer adaptation is performed. Other research attempts to determine less noise-sensitive features. One such approach is the use of root-cepstrum coefficients (RCC). They have been shown to be more immune to noise and lead to faster decoding [4]. Root cepstrum coefficients differ from the more standard Mel-frequency cepstrum coefficients in that the square root is taken rather than the logarithm. Matched training is yet another approach used to obtain noise robust features. While perfect performance has not yet been achieved, these methods are generally more computationally efficient than model adaptation, and often perform better. It is thought that if acoustic mismatch between trained models and test data could be completely eliminated, ASR performance would improve, even for low signal-to-noise ratio (SNR) levels.

Speech signal processing algorithms that extract speech characteristics, usually called features, are referred as the front-end of an ASR system. Front-end processing is a critical component of a speech recognition system. However information extracted in the front-end processing is unable to capture all necessary speech characteristics for accurate classification. There are currently no feature extraction methods that can fully separate every category of speech. Therefore some form of higher-level speech modeling is necessary. This higher level modeling is normally accomplished with either a Neural Network (NN) and/or a Hidden Markov Model (HMM). This stage of automatic speech recognition is referred to as the back-end. Because of its ability to model variability in speech durations, the Hidden Markov Model has been a primary

choice for the back-end of statistical speech recognizers for over 20 years. Hidden Markov Models do have certain limitations. One is that they are unable to model temporal information within states. For the sake of tractability, the assumption is made that the probability of the current symbol being generated is independent of any previously generated symbols. This assumption of independence is not strictly correct [14], [58], but does hold approximately. It has been shown that the inclusion of temporal information greatly improves the performance of automatic speech recognition systems [5], [6], [7].

1.2 RELATED WORK

This work focuses on improving feature extraction via Linear Predictive Coding with morphological filtering. Although Linear Predictive coding methods have previously been implemented in speech recognition this approach incorporates smoothing of the LP spectral envelope via new morphological filtering. Using Linear Predictive Analysis, static information is extracted from the acoustic sample over short time intervals on the order of 35ms. The envelope is smoothed with morphological filtering. Dynamic, or temporal, information is then obtained from the resulting feature vector containing static speech information.

Some researchers believe that more useful speech information is contained within the peaks of the envelope especially when noise or distortion is present. Peak detection is also an important component of pitch tracking algorithms. Thus, a significant amount of effort has been spent trying to improve ASR by emphasizing peaks and de-emphasizing valleys in the envelope. Speech information that has been overwhelmed by noise is often referred to as missing

data. Two approaches to achieving noise robust speech recognition with missing data have recently been under investigation. One approach has been to improve Hidden Markov Models via model adaptation to ignore missing speech information. Another has been to modify the front-end analyzer to better capture the speech information by ignoring speech segments overwhelmed by noise. Lippman and Carlson showed that, when channel variability and noisy conditions exist, missing feature compensation can be used to adapt the recognizer to reduce mismatch between training and test conditions and improve the accuracy of automatic speech recognition [8]. Missing feature compensation dynamically modifies the probability computations of HMM recognizers to better represent corrupted utterances. Even with severe environmental variability, missing feature adaptation provides high-performance speech recognition. Two components of missing data techniques can be applied to traditional ASR systems. Identifying corrupted portions of the speech spectrum is the first task. The second is modifying the ASR models to account for the missing signal components [9]. Additionally, significant performance improvements have been obtained in simulated noisy environments using a bounded marginalization approach for incorporating reliability evidence in frequency bins, with the potential for greater performance gains through better estimates of frequency dependent background noise levels [10]. Currently none of these methods has achieved the ideal performance.

1.3 SCOPE OF DISSERTATION

This dissertation investigates the use of spectral peak enhancement and peak envelope smoothing as an attempt to reduce feature variability, thereby achieving noise robust features for improved ASR performance. Specifically, spectral peak enhancement via LP analysis with discrete cosine transformation, followed by spectral envelope smoothing via morphological was developed and applied to the Aurora 2.0 and Aurora 3.0 speech databases. The idea of the morphological filtering is to use the shape-based organization of morphology and to expand the morphological filtering operator beyond the standard maximum and minimum operators. An important property of morphological filters is their resistance to outlying values and impulsive noise. With respect to this property, our goal was to suppress noise without introducing deterministic bias that would adversely affect recognition performance. Morphological filtering is expected to work well and be shown appropriate for speech signal enhancement.

The method was implemented with matched and mismatched training and the new features were compared with results from established techniques for each modality. Linear prediction analysis was used to obtain the first spectral representation of the speech signal. The LP spectral envelope was then smoothed with morphological filtering. Morphological filtering was used to enhance the peak regions of the envelope and reduce the valleys, where noise is presumed to predominate. DCTC features extracted from a DCT of the log magnitude spectrum were used as static features. A Hidden Markov Model framework with standard training and recognition algorithms was used

throughout to statistically model the speech. A second, more complex HMM framework was also used and compared to the first statistical speech model. The primary objective of this work was to investigate spectral envelope smoothing and peak enhancement for the purpose of improving ASR. Placing this work in a mathematical framework such as finding and removing corrupted speech segments and minimizing a global mean square error representation is normally too general and may or may not lead to improving ASR accuracy. Ultimately, it is system performance in actual environments that is critical. Therefore, the goal of this dissertation is to contribute some knowledge in the field which could later be added to achieve additional performance improvements. The specific objectives of this work are:

- Develop a morphological filter structure encompassing linear and nonlinear operations for envelope smoothing, which helps to overcome the deterministic bias problems of standard morphological filtering.

- To provide analyses of the behavior of the new morphological filter, and illustration of its success as regards noise reduction for automatic speech recognition systems.

- Integration of morphologically based spectral envelope smoothing, and linear predictive coding front-end processing algorithms that yield a set of noise robust features and improved automatic speech recognition.

1.4 OUTLINE OF DISSERTATION

Chapter 2 contains a brief review of digital signal processing for automatic speech recognition, including filter bank analysis, computation of Mel-frequency cepstral coefficients (MFCC), and comparison of discrete cosine transform series

coefficients with subsequent discrete cosine series (DCTC-DCS) based cepstral coefficients with MFCCs. This comparison points out the benefits of DCTC-DCS coefficients for automatic speech recognition and motivates additional processing to enhance the speech signal before computing statistical models for automatic speech recognition. Feature computation techniques based on linear predictive analysis with subsequent envelope smoothing via morphological filtering length are also presented. Chapter 3 discusses the Aurora Distributed Speech Recognition (DSR) task and the Aurora 2.0 & 3.0 databases. Basic concepts of HMMs are summarized in Chapter 4. The Hidden Markov Tool Kit (HTK), used to implement HMMS for speech recognition reported in this work, is included in the discussion of HMMs. Chapter 5 describes the control experiment defined and reported on by the Aurora working group. A second control experiment using the Old Dominion University Speech Communications Laboratory FFT-derived DCTC/DCS signal model, and spectral modeling with LP-derived DCTC/DCS coefficients are also discussed in this chapter. Morphological filtering is introduced and discussed in Chapter 6. Subsequently, spectral envelope smoothing via Morphological Filtering is presented, followed by the experiments conducted using proposed morphological filtering of signal models. Finally, Chapter 7 presents conclusions of the dissertation and suggestions for future work.

CHAPTER II
SPEECH SIGNAL REPRESENTATION

2.1 INTRODUCTION

In order to implement an automatic speech recognizer, the analog speech signal must be sampled and converted to its digital representation. This first stage is usually handled by the system hardware, i.e. an analog to digital signal processing board. The speech data used in this work was created from the TI-digits database, which was "collected at TI in 1982 in a quiet acoustic enclosure using an Electro-Voice RE-16 Dynamic Cardiod microphone, digitized at 20 kHz." [54], using 12 bit quantization for the analog to digital (A/D) conversion [86]. Although the (A/D) conversion process can have a significant impact on the performance of an ASR system only existing databases were used. Therefore, no new A/D techniques were employed during the course of this research. Thus, the analog-to-digital conversion process is not addressed in this dissertation. The second stage is to parameterize the digital signal to obtain a parametric representation that emulates observed human auditory and perceptual systems.

Speech signal modeling is the process of converting digital sequences of speech samples to observation vectors. Unfortunately, speech signals contain a significant amount of information that is not useful for automatic speech recognition. Therefore, speech signal modeling algorithms are designed to parameterize salient spectral energies of the speech sample, which are then helpful in maximizing automatic speech recognition performance, while minimizing the dimensionality of the speech representation vectors. Additionally,

signal modeling attempts to achieve parameterizations that are robust to noise as well as speaker and channel variations. The final requirement is to capture spectral dynamics in the signal parameterization.

Speech information is assumed to be the convolution of the input, or excitation, and the vocal tract or impulse response [14]. For speech recognizers the vocal tract imparts more useful information than the excitation source. A primary goal then is to separate and preserve vocal tract information for the determination of the fundamental units of speech while ignoring the effects of speaker differences, channel distortion, and background noise. Thus, in order to analyze the speech signal it must be de-convolved.

Analysis of a speech signal naturally includes de-convolution of the source and signal. Along those lines there are two approaches to signal modeling. Articulation-based signal representations for speech recognizers that attempt to model speech signal properties that reflect the shape of the vocal tract, rather than the excitation source, comprise the first approach. The second approach uses perceptually-based signal representations in an attempt to model the frequency response of the human ear, which is essentially insensitive to phase effects. In both cases a signal representation that ignores short time phase effects is desirable. One example that ignores short time phase effects is the short-time power spectrum. In the log-power domain the source and vocal tract become additive components and are therefore easier to separate. An important property of the log power spectrum is that its shape is invariant to gain applied to the speech signal. The spectrum is translated up or down by the applied gain,

but not distorted. Another important characteristic of the log-power spectral representation is that effects of channel distortion from communications channels are additive constants in the log-power domain. In the time domain these components are multiplicative. Additive noise presents a more challenging problem.

Extraction and preservation of speech information is critical but a speech recognizer also needs to be robust in its ability to ignore the effects of background noise, channel distortion, and speaker differences. Additionally, speech signal representations need to be as compact as possible so that large data sets can be processed in reasonably short time periods, with as little computational demand as possible. Many efficient representation methods use short-time time or spectral analysis methods to model either the function of the human ear or the vocal tract. Although only Linear Prediction Analysis and Global Shape Analysis (DCTC/DCS) are investigated in this work, some of the historically important modeling techniques for speech are described below. For further reading on this subject refer to "Signal Modeling Techniques in Speech Recognition," [15], a readable yet comprehensive introduction to this topic.

The remainder of this chapter is organized as follows. Historically important signal modeling techniques used in speech processing are described in Section 2.2. Section 2.3 presents The Old Dominion University modeling technique called Global Spectral Shape Analysis. Section 2.4 provides an overview of speech signal peak detection, and spectral envelope smoothing is discussed in Section 2.5. Chapter conclusions are given in Section 2.6

2.2 SIGNAL MODELING TECHNIQUES: AN OVERVIEW

Although numerous modeling techniques have been attempted, with varying degree of success, only the signal representation methods with the most historic significance are presented in this overview. Thus, this section provides a brief discussion of Filter bank analysis, Mel-frequency Cepstral Analysis, and Linear Predictive Analysis.

2.2.1 FILTER BANK ANALYSIS

One of the first speech signal representations used for ASR was implemented with a filter bank. The filter bank was originally constructed as a set of overlapping triangular shaped band-pass filters which typically covered a frequency range of 0 Hz to 5 kHz. This was originally accomplished with analog circuitry. However, the availability of desktop computers and efficient digital signal processing has made it much easier to realize filter banks with software. Additionally, software modeling is easier, and less expensive to adapt to changing demands. Filter banks are designed to model the nonlinear frequency perception of the human ear. As a result they fall into the class of perceptually-based speech modeling methods. Filter bank spacing is commonly determined using the Mel scale, proposed by Stevens, Volkman, and Newman in 1937, to emulate the nonlinear frequency perception of the human ear.

The Mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another [16]. Another psycho-acoustical (perceptual) scale, called the Bark scale, ranges from 1 to 24 Barks. The barks correspond to the critical band of human hearing, and range from 0 Hz to 15.5 kHz.

2.2.2 MEL-FREQUENCY CEPSTRAL ANALYSIS

Mel-Frequency Cepstral Coefficients (MFCCs) are perceptual-motivated coefficients derived from the Inverse Fourier Transform of the log spectrum. The MFCCs approximate the human auditory response by logarithmically positioning frequency bands on the Mel scale, rather than the linearly spaced frequency bands obtained with the Fourier Transform. MFCCs are computed, on a frame by frame basis, as follows: The Fourier Transform of overlapping speech segments is computed. Next the transformed signal is squared to obtain the spectral magnitude, and then summed to obtain filter outputs. Frequency bands are formed by grouping neighboring coefficients. This grouping also induces the Mel frequency scale and reduces fine harmonic structure at multiples of the fundamental frequency, F_0.

The Discrete Cosine Transform (DCT) of the log energy of each filter reduces correlation in adjacent energy levels, induced by the vocal tract, and gives the final Mel Frequency Cepstral Coefficients. MFCCs preserve important information while reducing the number of required coefficients for subsequent statistical modeling. The first cepstral coefficient is a measurement of the shape of the log spectrum and is called C_0. The second coefficient, C_1, gives a measure of the balance between the two halves of the spectrum. Higher order coefficients provide information regarding the finer features of the spectrum.

2.2.3 LINEAR PREDICTION ANALYSIS

The human vocal tract can be modeled as a lossless acoustic tube with plane-wave sound propagation along the tube. As a result, the impact of the

vocal tract on the excitation signal is that of a series of resonances [17]. Consequently the vocal tract can be roughly modeled as an all-pole filter. Linear prediction (LP), also known as auto-regressive (AR) modeling, is a least mean squared error algorithm that fits the parameters of an all-pole filter to the speech spectrum. The estimated speech signal is determined by a linear combination of past samples. The relationship is given by the following equation,

$$\hat{x}[n] = \sum_{k=1}^{p} a_k x[n-k], \tag{2.1}$$

where $\hat{x}[n]$ is the estimated speech signal and $x[n-k]$ are the past samples. Using Z-transform notation the transfer function of the linear prediction filter is obtained from equation 2.1 and is given in equation 2.2.

$$H(z) = \frac{G}{\sum_{k=0}^{p} a_k z^{-i}}, \tag{2.2}$$

where G is the gain, a_0 = 1, p is the number of poles and is referred to as the order of the LP analysis. Spectral peaks are located at the roots of the denominator. From equations 2.1 and 2.2 it can be seen that LP analysis predicts the current sample as a linear combination of the past p samples. Choosing the order of LP analysis can be difficult and, in many cases, is empirically determined. Generally higher order results in lower prediction errors. Unfortunately, when the order becomes too large the model fits individual harmonics and the separation of the vocal tract and excitation is not very good.

In equations 2.1 and 2.2 the $\{a_k\}$ are filter coefficients selected to minimize the mean squared error over the analysis frame. Short time analysis is

used to estimate the coefficients. For each segment the squared error is computed and minimized by setting the derivative equal to zero. We can express the error for each segment as a set of p linear equations as follows:

$$E_n = \sum_{m=0}^{N-1} e_n^2[m] = \sum_{m=0}^{N-1} (x_n[m] - \tilde{x}_m)^2 = \sum_{m=0}^{N-1} (x_n[m] - \sum_{k=1}^{p} a_k x_n[m-k])^2, \quad (2.3)$$

where N is the length of the analysis frame. Minimizing the mean squared error results in the Yule-Walker equations:

$$\sum_{k=1}^{p} a_k \varphi_m[i,k] = \varphi_m[i,0], \quad (2.4)$$

where the φ_m are covariance functions and the a_k are the prediction coefficients.

There are numerous methods for solving the Yule-Walker matrix equations. However, due to the symmetric windowing of the speech samples, and the resulting short time nature of the speech segments or frames, the autocorrelation matrix is Hermitian and has the Toeplitz property. Thus the parameters can be estimated using Levinson-Durbin recursion. Since this method is always stable, it has been become one of the most common approaches for speech recognition. The Covariance method using Cholesky matrix decomposition has also been used to compute the filter coefficients. It introduces the least amount of bias but the method does not always result in stable filters. Therefore, it is not commonly used in speech recognition. The Covariance method is also known to reduce the amplitude of spectral peaks which is contradictory to the goal of emphasizing peaks in noisy speech. Thus, the method was not a good candidate for this dissertation work. Another

alternative is the Lattice method, which is equivalent to Levinson-Durbin recursion. This method can be highly computationally demanding but is sometimes preferable because it also results in stable filters.

2.3 GLOBAL SPECTRAL SHAPE ANALYSIS: DCS ANALYSIS

Historically, ASR systems have relied on FFT-based spectral analysis. The FFT of the speech signal is taken and filter bank analysis is applied to the compute Mel-frequency Cepstrum (MFC) and Mel-frequency Cepstral Coefficients (MFCC). The Old Dominion University Speech Communication Lab uses a technique for feature extraction based on the encoding of global spectral shape [18], [19], [20]. While MFCC analysis applies the DCT to the energy of the spectrum across all filter bands, the method used in this research applies the DCT directly to the log magnitude spectrum. We refer to the resulting features as Discrete Cosine Transform Coefficients (DCTCs). A modified DCT is used to model the non-linearity of the human ear in speech perception. This modification is achieved via a bilinear warping of the basis vectors of the standard DCT [18]. The effect achieved by the warped basis vectors is more resolution at low frequency and less resolution at high frequency. This concept is similar to the non-uniform distribution of the filters in Mel-scale filter bank analysis, and emphasizes the lower frequency regions of the speech signal while de-emphasizing the higher frequency regions.

It has been shown that the addition of spectral-temporal information derived from time derivatives can greatly improve the performance of automatic speech recognition systems [5]-[8]. The objective is to capture the temporal

changes of each feature component from frame to frame. Vector components containing spectral-temporal information from time derivatives are commonly referred to as dynamic features. Several adjacent frames are used to extract this dynamic information. First order time derivative components, D_t, are called delta coefficients, where t represents the signal sample at time t. They can be computed with the following regression formula

$$D_t = \frac{\sum_{i=-W}^{W} iC_{t+i}}{\sum_{i=-W}^{W} i^2}, \qquad (2.5)$$

C_t are the corresponding static coefficients and W is the time span, in number of frames, in each direction around the center time t.

Second order time derivatives, or acceleration coefficients are computed using equation 2.5 with the delta coefficients; in some cases the window size for delta-delta coefficients is different than the window size for delta coefficients. A window size of 3 is frequently implemented [21], but there are systems that use a much longer window size [22], [23]. Since the delta and acceleration coefficients are concatenated to the static feature vectors the resulting vectors are higher in dimensionality than the static feature vectors. For this reason it is sometimes necessary to perform vector transformation to reduce the dimension of the feature space before proceeding with statistical modeling.

At The Old Dominion University Speech Communications Laboratory the spectral-temporal features are determined using a Discrete Cosine Series Expansion over time. Actually the DCT is applied over a block of frames each of

which represents one time instant. The resulting parameters are called Discrete Cosine Series Coefficients (DCSCs). An important difference between DCSCs and MFCCs with delta and delta-delta terms is that the first DCSC is the smoothed version of the corresponding DCTC, resulting in a final signal representation that is more noise robust but which does not represent the fine detail of the speech signal.

Zahorian and Nossair first presented the following equations [18] to show the derivation of DCTC and DCSC parameters. For completeness the details are provided here. First, let $X(f)$ be the magnitude squared spectrum represented with linear amplitude and frequency scales and let $X'(f)$ be the magnitude spectrum as represented with perceptual amplitude and frequency scales. Let the relations between linear frequency and perceptual frequency, and linear amplitude and perceptual amplitude, be given by:

$$f' = g(f), \quad df' = \frac{dg}{df} df, \quad X' = log(X). \tag{2.6}$$

For convenience of notation in later equations, f and f′ are also normalized to the range [0,1]. The acoustic features for encoding the perceptual spectrum are computed using a cosine transform,

$$\{DCTC\}_i = \int_o^1 X'(f') \cos(\pi i f') df', \tag{2.7}$$

where $\{DCTC\}_i$ is the i-th feature as computed from a single spectral frame. Equation 2.8 can be obtained from equation 2.7 by substituting equation 2.6 into equation 2.7.

$$\{DCTC\}_i = \int_0^1 log(X(f))\, cos[\pi i g(f)]\, \frac{dg}{df}\, df. \qquad (2.8)$$

We therefore define modified basis vectors as

$$\phi_i(f) = cos[\pi i g(f)]\, \frac{dg}{df}, \qquad (2.9)$$

and rewrite the equation as follows:

$$\{DCTC\}_i = \int_0^1 log(X(f))\, \phi_i(f)\, df. \qquad (2.10)$$

Thus, using the modified basis vectors, all integrations are with respect to linear frequency. In practice, equation 2.10 can be implemented directly as a sum using the squared spectral magnitude of the FFT. Any differentiable warping function can be precisely implemented, eliminating the need for the triangular filter bank. The DCTC terms computed with equation 2.10 are very similar to cepstral coefficients. However, to emphasize the underlying cosine basis vectors and the calculation differences relative to most cepstral coefficient computations, we call them the Discrete Cosine Transform Coefficients (DCTCs). This is consistent with terminology in previous related work [17] [20].

In this dissertation, DCTC parameters were computed with equation 2.9 using a logarithmic amplitude scale and the bilinear warping function given in equation 2.11, with warping coefficient α.

$$f' = f + \frac{1}{\pi} tan^{-1}\left\{\frac{\alpha\, sin(2\pi f)}{1 - \alpha\, cos(2\pi f)}\right\}. \qquad (2.11)$$

The value of α is chosen in order to emphasize the lower frequency region, where more speech information is present, and to de-emphasize the higher frequency region where there is less speech information.

The first three basis vectors, incorporating the bilinear warping, are shown in Figure 1. DCSC features were computed so as to encode the trajectory of the short-time spectra. Using the processing as described above, P DCTCs (P=13 is used in this work) were computed for equally-spaced frames of data spanning a segment of each token. Each DCTC trajectory was then represented by the coefficients in a modified cosine expansion over the segment interval.

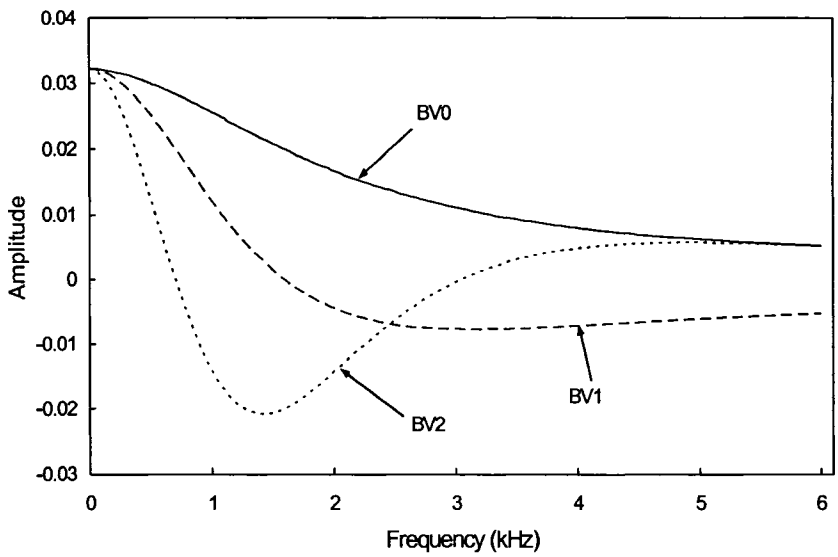

Figure 1 The first three DCTC basis vectors, with a warping factor of 0.45 (Zahorian and Nossair) [18].

The equations for this expansion, which are of the same form as equation 2.8, allow non-uniform time resolution as follows.

Let the relation between linear and perceptual time be given by

$$t' = h(t), \tag{2.12}$$

where $h(t)$ is a Kaiser window, chosen such that its derivative, dh/dt determines the resolution for t'. For convenience, t and t' are again normalized to the range [0,1]. The spectral feature trajectories are encoded as a cosine transform over time using

$$\{\{DCTC\}_i\}_j = \int_0^1 DCTC'(i,t')\cos(pjt')\,dt'. \tag{2.13}$$

The DCSC(i,j) terms in this equation are taken as the new features which represent both spectral and temporal information over a speech segment. Making the substitutions

$$\begin{aligned} t' &= h(t), \\ DCTC'(i,t') &= DCTC(i,t)' \\ dt' &= \frac{dh}{dt}\,dt \end{aligned} \tag{2.14}$$

equation (2.13) can be rewritten as

$$\{\{DCTC\}_i\}_j = \int_0^1 DCTC(i,t)\cos[\pi jh(t)]\,\frac{dh}{dt}\,dt. \tag{2.15}$$

We again define modified basis vectors as

$$\theta_j(t) = \cos[\pi jh(t)]\,\frac{dh}{dt} \tag{2.16}$$

and rewrite equation 2.15 as

$$\{\{DCTC\}_i\}_j = \int_0^1 DCTC(i,t)\,\theta_j(t)\,dt. \tag{2.18}$$

Using these modified basis vectors, feature trajectories can be represented using the static feature values for each frame, but with varying resolution over a segment consisting of several frames. To emphasize the underlying cosine basis vectors and to differentiate between expansions over time (DCSC) versus

expansions over frequency DCTC the terms computed in equation 2.18 are referred to as Discrete Cosine Series Coefficients (DCSCs). In general, each $DCTC_i$ was represented by a multi-term $DCSC_j$ expansion.

By varying the Kaiser Beta parameter for $h(t)$, the resolution can be changed from uniform over the entire interval (beta = 0) to much higher resolution at the center of the interval than the endpoints (beta values of 5 to 15). Figure 2 depicts the first three DCSC basis vectors, using a coefficient of 5 for the Kaiser warping function. The motivation for these features is to compactly represent both spectral and temporal information with considerable data reduction relative to the original features. For example, if 4 DCSC basis vectors are used for each expansion, then 12 DCTCs computed for each of 50 frames (600 total features) can be reduced to 48 features.

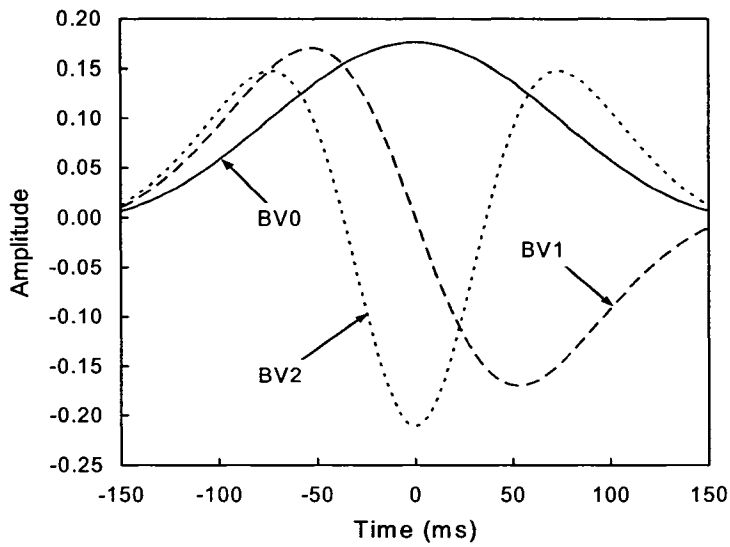

Figure 2 The first three DCSC basis vectors, with a coefficient of 5 for the Kaiser warping function (Zahorian and Nossair) [17].

2.4 PEAK DETECTION

The all-pole assumption of LP analysis is essentially invalid and real speech is not periodic. However, the non periodic nature of speech is frequently addressed by the use of short time analysis, which allows quasi-periodicity of speech signal to be assumed. The all-pole assumption does not completely represent real speech because the human speech production system has many branches. For example, the nasal cavity and the mouth cavity are both part of the total speech production system, and are connected to the vocal tract. The opening and closing of the velum can change the tone of the produced speech.

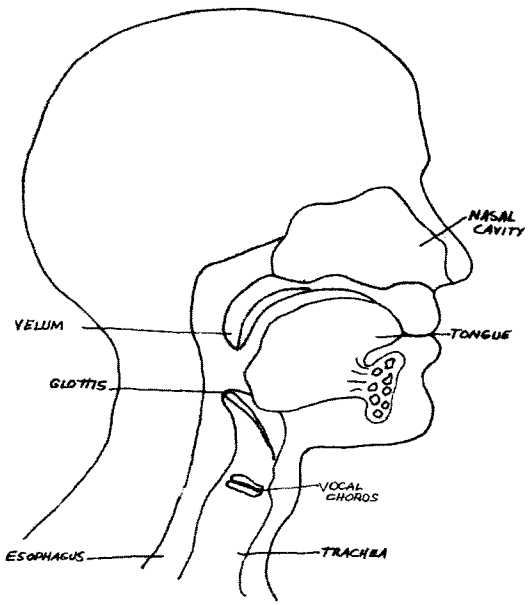

Figure 3: The human speech production system.

The zeros introduced by such branches are clearly not represented by the all-pole representation. However, the model has been shown to give a signal representation method which results in good Automatic Speech Recognition

performance. Figure 3 provides a view of the major parts of the human speech production system [83].

Part of this research involves the hypothesis that by using pitch synchronous analysis, we can potentially obtain further improvements in machine recognition performance. Phonemes are sometimes distinguished by fundamental frequency properties. Thus, the fundamental frequency itself can be an important parameter to improve recognition accuracy. Additionally, pitch characteristics are the most widely considered acoustic characteristics for stress evaluation [23]. As a result, accurate, robust estimation of the fundamental frequency, F_0, has long been an important problem in speech processing. In 1976 Rabiner proposed spectral peak extraction for fundamental frequency estimation [24]. More recently, De Chaveigne [25], Shimamura [26], and others [27]-[38] have proposed methods for the use of spectral peaks in improving recognition accuracy and/or increasing noise robustness. In this dissertation a form of peak detection is implemented as a means to improve spectral envelope smoothing as discussed in the next section. The motivation for the use of peak detection in envelope smoothing came from previous work in pitch tracking. Therefore, a brief overview of fundamental frequency estimation is given here.

In speech recognition pitch trackers are generally implemented in an attempt to estimate the fundamental frequency, which is an inherent property of periodic signals. Of course speech is not periodic but in short-time analysis short enough segments are taken so that quasi-periodicity can reasonably be assumed for each segment. To be more precise, pitch itself is a perceptual phenomenon,

a nonlinear function of the temporal and spectral energy distribution of the sound, and is not directly measurable from the speech signal. To be rigorously correct, the term pitch should be used to refer to the auditory perception of tone [28]. However, it is then important to note that pitch trackers actually attempt to track F_0 because it approximately correlates with perceived pitch [29].

In the time domain the fundamental frequency is frequently estimated via auto-correlation of the input signal. Spectral methods of F_0 estimation rely more often on spectral peak detection. Thus, in the spectral domain a pitch tracker is essentially searching for the smallest true period, T_p, for each short time interval analyzed. Upon determination of this smallest period the fundamental can be taken as $\frac{1}{T_p}$. Estimation of the fundamental normally includes three primary phases: signal conditioning or pre-processing to remove noise and DC offset, estimation of candidates, and post-processing to select the best candidates for the fundamental, F_0. Naturally, the second and third stages depend heavily on the location of spectral peaks.

The ability to locate the fundamental frequency is influenced by many factors, such as low frequency, and DC components in the speech signal. As a result, each of the three components mentioned in the previous paragraph must be designed to contend with a large degree of sensitivity in the parameter being estimated. The reason for this is that in order to produce the rich variety in human speech sounds the vocal tract takes on a huge range of shapes. As mentioned earlier, there are many components of the vocal tract. For example,

two other components that contribute to deviation in the fundamental frequency are the human glottis, which can exhibit time-dependent chaotic behavior [85], and vocal chords covered in a mucus membrane that frequently redistributes itself [29]. Other factors that increase the difficulty of estimation of F_0 are spectral distortion caused by microphones and telephone handsets which can completely remove the fundamental frequency and background noise that in some cases overwhelm the fundamental frequency. For more information regarding the human sound production system the reader is referred to [9], [14], [16].

2.5 ENVELOPE SMOOTHING

Although the objectives are different, signal representation is important for both speech analysis and speech synthesis. In speech synthesis the goal is to produce natural speech sounds from a given representation. As with speech recognition the signal representation has a significant impact on the quality of the final system output. As already stated, the speech recognition goal in this dissertation is to find a speech signal representation that can be used to improve machine recognition of noisy speech.

Researchers interested in speech synthesis have traditionally used signal representations similar to those used for speech analysis and recognition. For example, envelope smoothing has recently been investigated for the purpose of reducing the perceptible buzz in synthesized speech [39]. Linear predictive coding is another one of the many methods that have been investigated for speech synthesis. It has been shown that random variations in spectral representations, which are due to local periodicity in the signal, can be observed

in LP models. Although human voiced sounds are perceived to be smoother than unvoiced sounds the local periodicity can cause errors in the estimation of the fundamental. Thus, a spectral representation with no periodicity is desirable and would allow for a method of F_0 estimation that provides a smooth trajectory.

The use of envelope smoothing in speech synthesis was a motivating factor in this work towards noise robust speech recognition via LP modeling with subsequent morphological envelope smoothing. Smoothing is implemented in the frequency domain rather than the time domain as is done in many of the speech synthesis systems. The LP parameterization provides a good spectral representation for addressing problems related to local periodicity. However, the LP model assumes the source of the periodic component is a regular pulse train [40]. This assumption is not valid for human speech because it represents only poles, also referred to as auto-regressive components. The moving average components, which would be represented by zeros, are not represented by the envelope of the LP model. Thus, the model is based on only partial information and is not completely representative of natural speech. The goal was to compensate for this by smoothing of the spectral envelope via morphological filtering. Since the signal is filtered on a frame-by-frame basis the smoothing is achieved on a local level.

The first attempt for envelope smoothing was to first explicitly locate spectral peaks. However, noisy speech has natural as well as induced fluctuations making spectral peak identification numerically fragile. In contrast, morphological filtering is a convolution operation based on dilation and opening

of the signal, and results in a stable approach to peak enhancement. The filtering can be implemented to emphasize and broaden peaks, but no explicit peak identification is required. Figure 4 illustrates the smoothed spectral peak envelope (the dotted line), obtained via a combination of LP analysis, morphological filtering, and signal thresholding. The solid line is the original spectrum. More details of this signal processing are given Chapter 6. In this dissertation morphological filtering is implemented in the front-end and introduces no significant loss of computing speed in the recognition phase. However, there is an increase in computational demand in the front-end algorithm which is an attempt to determine a signal representation that is robust to additive and convolutional noise.

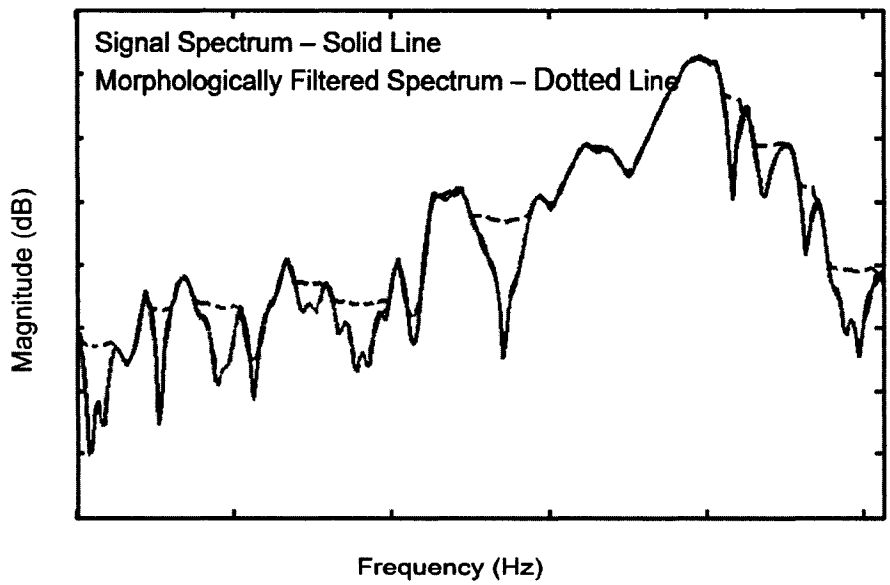

Figure 4 Illustration of envelope smoothing via LP analysis and morphological filtering.

2.6 CHAPTER CONCLUSION

This chapter gave a brief overview of several historically important signal modeling techniques used in speech processing, and presented the Global Spectral Shape Analysis developed for speech signal representation by the Old Dominion University Speech Communications Laboratory. Speech signal peak detection and spectral envelope smoothing were discussed as motivating factors for the work presented in this dissertation. Experiments presented in Chapters 5 and 6 support the hypotheses that enhancement of spectral peaks via LP spectral modeling and morphological filtering improves Automatic Speech Recognition performance in noisy conditions.

CHAPTER III
TASK AND DATABASE

3.1 INTRODUCTION

Historically speech recognition has taken place in a closed system where speech analysis and recognition are performed on the same machine. In Distributed Speech Recognition (DSR) collection, digitization, and parameterization of the speech sample is performed in a handset, either fixed or mobile, and the recognition phase is accomplished in a central location in a telecommunications network. As a result, the ETSI STQ-AURORA DSR Working Group [59] has been developing standards for Distributed Speech Recognition (DSR). Noisex-92 was the original database used in the evaluation of Distributed Speech Recognition systems but its use is restricted to isolated word recognition tasks. Therefore, there was a need for a database designed so that the performance of recognition algorithms for continuous speech in noisy conditions could be evaluated, and results from various research groups compared. The Aurora 2.0 database was the first database designed to meet this requirement and was contributed by the DSR working group [57]. This database can be obtained through the Evaluations and Language Resources Distribution Agency (ELDA).

The remainder of this chapter is organized as follows. Section 3.2 provides a detailed description of the signal processing used to create the Aurora 2.0 database. This section also presents the organization of the Aurora 2.0 training and test data, and finally gives a listing of experimental results obtained

using the database, published by the ETSI working group. Section 3.3 gives a similar description of the Aurora 3.0 database along with results published by the respective groups who created the individual databases which make up the Aurora 3.0 database. A brief description of some key studies using Aurora 2.0 and 3.0 are also provided in Section 3.4. This is followed by results published by the research groups who conducted the key studies. Chapter conclusions are discussed in section 3.5.

3.2 AURORA 2.0 DATABASE

Aurora 2.0 waveforms were created from the TI-digits database by digitally adding 8 different real-world background noises to 8440 utterances of connected digit strings taken from the TI-digits database. Each utterance was spoken by adult male and female American English speakers and consists of digit strings ranging from one to seven digits.

The original TI-digits database was recorded at 20,000 samples per second. In order to simulate band limited speech signals in a telecommunications system the original TI-digits database was down-sampled to 8,000 samples per second after filtering with a low pass filter with a bandwidth of 0 to 4,000 Hz. Artificial noise was added after the low-pass filtering. After down-sampling the digit sequences were filtered with a G.712 filter, defined by the International Telecommunications Union (ITU), to simulate frequency characteristics of telecommunications handsets and equipment [60]. The frequency response of the G.712 filter is given in Figure 5, which was taken from the ITU Recommendation G.712, "Transmission performance Characteristics of

Pulse Code Modulation Channels" [60]. For simulating the behavior of a telecommunication terminal the ITU defined the MIRS filter with the frequency response depicted in Figure 6, which was taken from the ETSI-SMG technical specification, "European Digital Cellular Telecommunication system GSM03.50 [71].

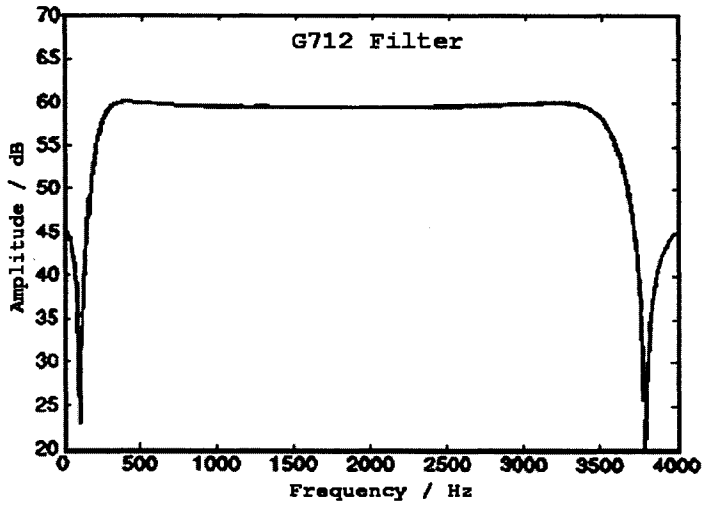

Figure 5 G.712 filter frequency response.

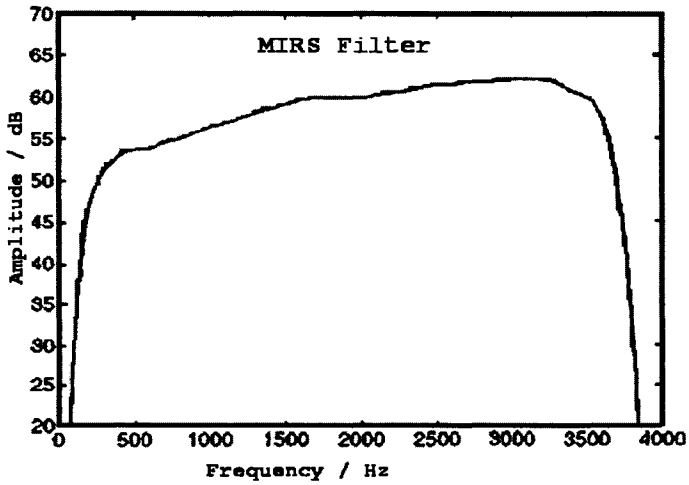

Figure 6 MIRS filter frequency response.

The purpose of the MIRS filter is to facilitate comparison of recognition performance when the frequency characteristic at the input differs [57]. Therefore, all of the training and test data were filtered with the G.712 filter while only one subset of test data was filtered with the MIRS filter.

Selection of noise types was determined by attempting to represent environments where the use of mobile handsets would be most likely. Thus, noises were recorded in the following types of locations:

- Suburban Train
- Babble
- Car
- Exhibition Hall
- Restaurant
- Street
- Airport
- Train Station

Each of the noise types was added to the down-sampled, filtered digit sequences at signal-to-noise (SNR) ratios ranging from -5 dB to 20 dB in steps of 5. This resulted in speech samples with seven different signal-to-noise ratios plus the original clean data.

3.2.1 TRAINING DATA

Training can be performed on clean data only or on clean and noisy data. When clean and noisy data are used for training the data is referred to as multi-condition data. To date multi-condition training has invariably yielded the best

performance when testing with noisy data matching the noise in the training data. It is thought that models derived from only clean training data contain no information relating to noise and channel distortion. This mismatch between the training and test data generally results in degraded performance. However, since it is often not possible to insure that noise conditions are matched between training and testing, results obtained using clean training data and noisy test data are of great interest.

Clean training data was produced by taking 8440 utterances from the training data in the TI-digits database and filtering them with the G.712 filter and ensuring that no noise was added. The Aurora 2.0 multi-condition training data was created using the same 8440 sentences taken from the TI-digits training data. These sentences were G.712 filtered, and split into subsets containing 422 sentences. The following four noise types were then added:

- Suburban Train
- Babble
- Car
- Exhibition Hall

Each of the noise types was added at signal-to-noise ratios ranging from 5 dB to 20 dB in steps of 5. For each noise type there is also a clean set that has not had noise added. The result of this procedure is the multi-condition training data consisting of 20 subsets, each containing 422 sentences.

3.2.2 TEST DATA

A total of 4004 utterances from the TI-digits test data were used to create the Aurora 2.0 test data. There were 52 male and 52 female speakers. First the utterances were split into 4 subsets of 1001 utterances per subset. Each subset contains recordings of each of the 104 speakers. As with the training data the entire set of test utterances was filtered with the G.712 filter, then one noise signal is added to each subset at signal-to-noise ratios ranging from -5 dB to 20 dB in steps of 5. Thus, for each of the four primary subsets there are six signal-to-noise ratios and one set of clean utterances.

Suburban train, babble, car, and exhibition hall noises were added to create Test Set A. These noise signals are the same as the noise added to the training data. Thus, the highly matched noise characteristics between the training data and this test set lead to higher recognition results with this test set than for the other two test sets. With seven noise levels and four noise types this test set has a total of 28,028 digit strings.

Restaurant, street, airport, and train station noise were added to create Test Set B. The added noise types differ from the noise types added to the training data, creating a mismatch in the noise characteristics of this test set and those of the training data. Test Set B also consists of a total of 28,028 test utterances.

Only suburban train and street noise were added to Test Set C. Thus, the test set contains only 14,014 test sentences. However, the utterances were filtered, before noise addition, with the MIRS filter.

3.2.3 ETSI RESULTS FOR AURORA 2.0 DATABASE

The results in Tables 1 and 2 were published by the ETSI working group that created the database [57]. Where recognizer performance was evaluated by its percent word accuracy, which is determined by the following equation:

$$\%Accuracy=(N-D-I)/N\times 100 \tag{3.1}$$

where N is the total number of number of labels in the reference transcriptions, D is the number of deletion errors, and I is the number of insertions [83].

These results are considered as the original "baseline" for Aurora 2.0, and were obtained using the WI007 front-end described in Chapter 5 Section 5.3.1 and the baseline HMM back-end, described Section 3.5. Also note that the average results were obtained with the following weight function,

$$W=0.4*TSA+0.35*TSB+0.25*TSC \tag{3.2}$$

where TSA, TSB, and TSC are the average word accuracies of the individual test sets A, B, and C. Each test set average is computed from the individual averages, excluding SNR -5, unless otherwise mentioned this method was used for reporting all results mentioned in this dissertation.

Table 1 ETSI Published Results for Aurora 2.0 Multi-Condition Training.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set	Average
Clean	98.54	98.54	98.56	98.54
SNR20	97.52	96.96	96.74	98.12
SNR 15	96.94	95.38	95.48	96.02
SNR 10	94.59	92.56	92.13	93.26
SNR 5	87.57	83.76	81.66	84.75
SNR 0	59.82	58.91	49.61	56.94
Average	89.16	87.69	85.70	87.77

Table 2 ETSI Published Results for Aurora 2.0 Clean Training.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
Clean	99.02	99.02	99.05	99.03
SNR20	95.25	92.77	94.29	94.14
SNR 15	87.33	81.33	87.84	85.36
SNR 10	67.70	59.00	74.16	66.27
SNR 5	39.47	31.92	50.24	39.52
SNR 0	16.95	13.69	24.16	17.61
Average	67.62	62.96	71.62	66.99

3.3 AURORA 3.0 DATABASE

Aurora 2.0 was created for evaluating front-end algorithms in simulated noise environments. As such, it does not capture all of the characteristics of a real noise environment. The Aurora 3.0 database was created in order to provide standardized data for testing algorithms in real noise. Aurora 3.0 is comprised of five subsets with each recorded in a different language: Spanish, Danish, Finnish, German, and Italian. Original recordings used to create Aurora 3.0 were taken from the respective Speech-dat database.

Each of the Aurora 3.0 subsets contains recordings from two types of microphones, close-talking (c0) and hands-free (c1). In each subset the same utterance was recorded by both a close-talking and hands-free microphone. Thus, each original recording contributed two recordings to the database. All five of the subsets are subdivided into the following three noise level categories, with all categories associated with automobiles:

- Low – town traffic with low speed rough road
- Quiet – stop with motor running
- High – High speed with good road condition

Automatic Speech Recognition performance has been evaluated with the Aurora 3.0 database using three standard mismatches of the noise conditions (Well, Medium and High). Well-matched testing used both types of microphones and all three types of noise for both training and test data. Medium-mismatch testing used training data from the hands-free microphone data with quiet and low noise conditions, and test data from the hands-free microphone with only high noise conditions. The high-mismatch test took training data from the close-talk microphone and all noise types, and test data from the hands-free microphone with low and high noise conditions. For each database the training data consists of 70% of each of the female and male speakers such that the utterances of these speakers represent approximately 70% of the utterances for each condition. Test data is comprised of the remaining 30% of the sentences.

Additionally, recordings in all subsets were conditioned with the following three steps:

- DC offset removal to convert from unsigned integers to signed integers.
- Down sampling from 16 kHz to 8 kHz using the ITU-T [72] software tools library
- Removal of speaker synchronization by an automated process.

Each of the Aurora 3.0 subsets was created and tested by a different research group and contributed to the Aurora DSR Working Group. Sections 3.3.1 through 3.3.4 provide a brief description of each of the subsets of the Aurora 3.0 Speech-dat database, and recognition results published by the respective research groups. The Italian database was not included in this dissertation. Therefore, the descriptions provided here do not include the Italian database.

3.3.1 SPANISH SPEECH-DAT

The Spanish SDC-Aurora database consists of 4914 recordings taken from the Spanish Speech-dat database, with the distribution of the utterances as follows:

- Quiet – 792 utterances
- Low noise – 2422 utterances
- High noise – 1700 utterances

These sentences are used in each of the well-matched, medium mismatch, and high mismatch experiment types. The number of sentences for each of the training and test data sets for each matching type is given below:

Match Type:	Well-Matched	Medium-Mismatch	High-Mismatch
Training data:			
	Quiet 532	Quiet 396	Quiet 266
	Low 1668	Low 1211	Low 834
	High 1192	High 0	High 596
Test data:			
	Quiet 260	Quiet 0	Quiet 0
	Low 754	Low 0	Low 377
	High 508	High 850	High 254

3.3.2 DANISH SPEECH-DAT

The Danish SDC-Aurora database consists of 4914 recordings taken from the Danish Speech-dat database, with the distribution of the utterances as follows:

- Quiet – 792 utterances
- Low noise – 2422 utterances
- High noise – 1700 utterances

These sentences are used in each of the well-matched, medium mismatch, and high mismatch experiment types. The data is distributed 70% for training and 30% for test data, as with the other databases.

3.3.3 FINNISH SPEECH-DAT

The Finnish SDC-Aurora database consists of 4399 recordings taken from the Finnish Speech-dat database. The sentences are also sub-divided into quiet, low, and high noise types. As with the other databases, the utterances are used in each of the well-matched, medium-mismatch, and high-mismatch experiment types, where the data is distributed 70% for training and 30% for test data.

3.3.4 GERMAN SPEECH-DAT

The German SDC-Aurora database consists of 4914 recordings taken from the German Speech-dat database. The sentences are also sub-divided into quiet, low, and high noise types. As with the other databases, the utterances are used in each of the well-matched, medium-mismatch, and high-mismatch experiment types. This data is also distributed 70% for training and 30% for test data.

3.3.5 ETSI RESULTS FOR EACH DATABASE

The results reported in tables 3 through 6 were published by the respective group that created the database.

Table 3 Word Accuracy for Spanish SDC Database.

Spanish Speech-dat Car Matching Condition	**Word Accuracy Adv DSR Front-end**
Well-matched	86.85%
Medium-mismatch	73.74%
High-mismatch	42.23%
Average	67.61%

Table 4 Word Accuracy for Danish SDC Database.

Danish Speech-dat Car Matching Condition	Word Accuracy
Well-matched	77.8%
Medium-mismatch	47.4%
High-mismatch	31.9%
Average	52.3%

Table 5 Word Accuracy for Finnish SDC Database.

Finnish Speech-dat Car Matching Condition	Word Accuracy Adv DSR Front-end
Well-matched	95.04%
Medium-mismatch	77.70%
High-mismatch	68.76%
Average	80.5

Table 6 Word Accuracy for German SDC Database.

German Speech-dat Car Matching Condition	Word Accuracy
Well-matched	90.58
Medium-mismatch	79.06
High-mismatch	74.28
Average	81.31%

3.4 KEY STUDIES USING AURORA 2.0 AND 3.0

Experimental results from several research groups are reported in Table 7. These results serve as comparisons for experimental results reported in this dissertation. The table presents results reported for the Aurora 2.0 database where column one re-states the ETSI average results reported in Table 1 and

columns two through five were reported by various research groups with the lead author listed in the respective column. Analysis details of each method can be found in [57], [62], [63], [66], [73]. Each set of results was obtained using front-end algorithms designed to deal with noisy speech signals. Each of the algorithms was evaluated with either the Aurora 2 or 3 databases. While the algorithms do not use Linear Predictive Analysis or Morphological filtering, they serve as a reasonable comparison because they were evaluated with the same data.

The results in column one were obtained with the baseline MFCC features from the ETSI WI007 front-end, with no additional signal processing. Features evaluated and reported in column two were based on variable frame rate, peak isolation, peak-valley ratio locking and harmonic demodulation; features evaluated in column three were obtained using stereo-based piecewise linear compensation for environments (SPLICE). Features evaluated in column four were obtained using signal-to-noise dependent waveform processing, and

Table 7 Key Study Results for Aurora 2.0 using Multi-Condition Training.

Author	**Aurora WI007**	**Cui, X.**	**Droppo, J.**	**Macho, D.**	**Chen, C.**
Test Set					
TSA	89.16	90.22	90.83	91.37	93.76
TSB	87.69	88.84	89.37	89.72	93.27
TSC	85.70	89.08	89.24	89.51	93.51
AVG	87.77	89.38	89.81	90.20	93.52

features in column five were computed by taking the mean and variance of the features and subsequently taking the auto-regressive moving average of the

means and variances. Although results reported in columns two through five were used as comparisons, it is important to note the significance of the additional signal processing used to obtain the reported performance improvements. Results reported in this dissertation were achieved with Morphological filtering as the single additional signal processing step. Table 8 presents results reported for Aurora 2.0 clean training data. Signal processing used to achieve these results were the same as those used to produce results reported in Table 7. Thus, these results were used as a comparison for results reported in this dissertation for evaluations with the Aurora 2.0 clean training data.

Table 8 Key Study Results for Aurora 2.0 Clean Training Data.

Author	**Aurora WI007**	**Evans, W.D.**	**Kim, H.K.**	**Cui, X.**	**Chen, C.**
Test Set					
TSA	67.62	76.01	81.26	85.48	87.58
TSB	62.96	72.60	82.6	85.77	88.41
TSC	71.62	79.16	83.07	84.10	87.05
AVG	66.99	75.61	82.18	85.27	87.74

Table 9 presents results reported for the Aurora 3.0 database where row one re-iterates the ETSI results reported in Table 1 and rows two through five were reported by various research groups with the lead author listed in the respective row. Analysis details of each method can be found in [65], [66], [68], [73]. As with Tables 7 and 8 each set of results were obtained using front-end algorithms designed to deal with noisy speech signals. Again, each of the algorithms was evaluated with the Aurora 3 database. Features evaluated in row

two were obtained using variable sampling frequencies and features evaluated in row three were obtained with the same method used in column five of Table 7. Row four contains results which were obtained with the same algorithm evaluated in column three of Table 7. Finally, features in row five were obtained with TRAPs features based on multi-band and multi-stream approaches.

Table 9 Key Study Results for Aurora 3.0 Database.

Author	Finnish	Spanish	German	Danish
ETSI Published	68.76	67.61	81.31	52.3
Bauerecker, H.	84.78	86.78	NA	74.44
Chen, C.	96.36	91.41	86.86	80.25
Droppo, J.	91.22	92.64	90.03	86.00
Andre, A.	91.42	94.48	92.88	94.57

3.5 HIDDEN MARKOV MODEL ARCHITECHTURE

The baseline HMM back-end

The baseline HMM architecture consisted of one 18 state word model for each digit, using a continuous density HMM with Gaussian probability density functions (referred to as mixtures) and a diagonal covariance matrix for each model. Specifically, there were 13 continuous density HMM models, one for each digit, an additional one to represent the "oh" pronunciation for zero, and two silence models. Each of the digit models was initialized with 18 states with 1 Gaussian mixture per state. Initial transition probabilities were all equal except for the first transition, which had transition probability of 1, and variances were all floored to 0.01. Additionally, only left-to-right transitions, and self-transitions were allowed. After 16 iterations of training the resulting word models each

contained 6 Gaussian mixtures, and the final silence model had 10 Gaussian mixtures.

Initialization of the word models was accomplished by segmenting the training utterances into equal lengths and estimating the global mean and variance for the entire training database. These global values were used to initialize each of the word models. The initial parameters were used to re-estimate the model parameters for each of the word models. Subsequently the silence model was initialized with 5 states, each with a single Gaussian mixture component. The second silence model was designed to model intra-word short pauses, and is called a short-pause model. It was created by adding transitions from states 2 to 4 and from states 4 to 2 in the silence model, then tying these transitions to state 3 in the silence model. Several more iterations of parameter re-estimation were performed, in which additional Gaussian mixtures were added to the word models and to the tied silence models.

When the final models were obtained, each test utterance was decoded using the Viterbi algorithm, which determines the model with the highest likelihood of matching each token in the test utterance. This standard HMM configuration is the same as that used in the published results for Aurora speech data.

3.6 CHAPTER CONCLUSIONS

This chapter began with a discussion of the reasons for the creation of the Aurora databases, followed by the detailed description of the creation and

organization of each database. Finally, Section 3.5 gave a detailed description of the standard Hidden Markov Model architecture used for each experiment in Chapter 5 and all but the final experiment in Chapter 6. The remaining chapters present signal processing for LP derived DCTC/DCS signal representation and envelope smoothing via Morphological filtering. Experimental results from Automatic Speech Recognition experiments using our signal representation methods will be included in the chapters where the signal processing is presented.

In order to provide a basis for comparison of performance of the methods presented in this dissertation, the latter part of this chapter presented a short list of key studies in which either the Aurora 2.0 or the Aurora 3.0 database were used for training and evaluation of the signal representation methods implemented by the research groups who conducted the key studies.

CHAPTER IV
STATISTICAL MODELING OF SPEECH PARAMETERS

4.1 INTRODUCTION

The determination of similarity and dissimilarity between speech patterns is the primary problem in speech recognition. Similarities are typically used for classification of input or test patterns based upon a set of reference patterns. In order to classify a given speech observation the characteristics of the reference (training) speech patterns must be described by a well-defined mathematical model. There are several approaches to the modeling of speech signal characteristics. The Hidden Markov Model approach assumes that the speech signal can be well characterized as a parametric random process, and that the method of estimation of the parameters of the stochastic process is well-defined. Another approach is non-parametric and is implemented via Artificial Neural Networks.

Template matching was one of the first methods used in speech recognition. A template matching based acoustic-phonetic recognizer attempts to classify every analysis frame according to a defined set of features, such as flatness, compactness, and stress. Decisions are based on the presence of acoustic characteristics of the sample speech segment. Some typical acoustical parameters are spectral energy, duration, and formant location. Other possible methods are Dynamic Time Warping and Acoustic Segment Modeling. Each modeling technique makes specific assumptions regarding the signal. Therefore,

the modeling method used to compare acoustic patterns must be highly dependent upon the particular speech recognition system to be implemented.

Artificial Neural Networks and Hidden Markov Models are currently the most widely used methods for ASR. Current acoustic-phonetic recognizers typically implement an Artificial Neural Network (ANN) classifier to apply a mathematical rule to make decisions regarding classification of given speech patterns. Artificial Neural Networks are considered to be discriminative classifiers. They generally classify reference patterns by attempting to partition the speech space. Speech patterns are determined by attempting to place them into equivalence classes within the partitioned space. Theoretically, each speech token will fit into a unique equivalence class, where a token is the base speech unit chosen for the recognition task (e.g. vowels and consonants). In reality, no one has been able to completely partition the space because speech signals contain overlapping pieces of information. For example, formants often overlap between various speech tokens. Because of its more robust capabilities the Hidden Markov Model Method was used in this research. However, a brief overview of Artificial Neural Networks will be given in the next section, followed by a more in depth discussion of Hidden Markov Models.

4.2 ARTFICIAL NEURAL NETWORKS

From the most basic perspective a neural network can be viewed as a set of decision functions, frequently called neurons, which map input feature vectors directly to a classification decision [14]. The general objective is to determine the dissimilarity or distance between a speech sample to be classified and a set of

reference patterns, also referred to as categories or classes. The most common neural network architecture is a feed forward Multilayer Perceptron (MLP).

As mentioned previously, each component of a Neural Network is called a neuron. Multilayer Perceptron Neural Networks consist of an input layer, one or more hidden layers, and an output layer as depicted in Figure 7.

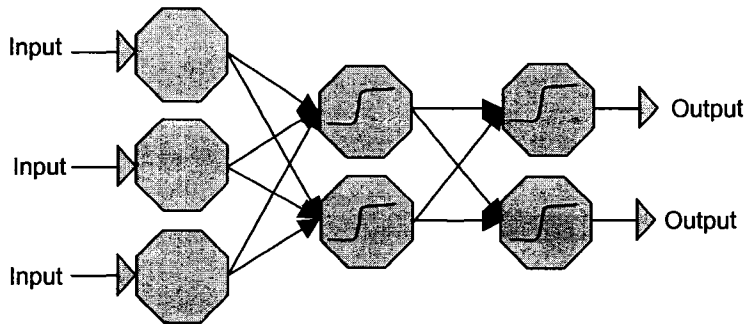

Figure 7 Artificial Neural Network.

The layers themselves contain a set of linear discriminant functions followed by nonlinear functions. The nonlinearity is typically a sigmoid defined by:

$$f(x) = \frac{1}{1+e^{-x}}, \tag{4.1}$$

where x is the input to the layer, and the linear discriminant functions are normally weight matrices. Training of an MLP involves adapting a weight matrix to associate an input with a target output. This process is achieved via the Error Back Propagation Algorithm using a steepest descent procedure that iteratively minimizes a cost function [41]–[47]. A Multilayer Perceptron with enough layers can be used to obtain an arbitrary mapping between the input and output layers.

Although Artificial Neural Networks are very fast at making correct classifications from unseen data, they do not provide high performance recognition with continuous speech. In contrast Hidden Markov Models have proven to yield significantly better recognition rates with continuous speech. In this dissertation all training and test data comes from the Aurora 2.0 and Aurora 3.0 databases which consist of connected digit strings, which makes them observations of continuous speech. As such the recognition task must be implemented using techniques which can effectively address the characteristics of continuous speech. Therefore, the decision was made to use Hidden Markov Models for statistical modeling rather than Artificial Neural Networks. A brief introduction to Hidden Markov Models is given in the following section.

4.3 HIDDEN MARKOV MODELS: INTRODUCTION

Markov Model Theory has been known for over 80 years but the theory was not useful for speech recognition until methods became available for parameter optimization. In the latter half of the 1960's such methods were proposed [58]. Hidden Markov Models were first implemented in speech science in the 1970's and 1980's [47]-[55], evolving from the search for effective statistical models of speech for speech synthesis. Hidden Markov Models address the stochastic behavior of the amplitudes of the feature vectors, and provide good characterization of the speech signal. They have proven to be quite useful because they generalize the pattern comparison process by determining statistical characterizations of spectral properties of the given reference categories, rather than making direct comparisons between reference

patterns and specific samples to be classified [56], [57]. Models produced with the Hidden Markov Modeling technique typically outperform other types of recognition systems because they determine the statistical parameters from the data, thus allowing them to capture more of the intra and inter-speaker variability that cannot easily be determined through other modeling techniques. Due to their stochastic nature, Hidden Markov Models are well suited for representation of a sequence of words or sounds. Thus, they are currently the first choice for recognition of continuous speech.

In signal modeling the non-stationary behavior of speech is addressed using short time segments such that each piece, or frame, can be considered stationary. In essence, the time varying nature of speech is viewed as a concatenation of the short time segments. Thus, there is an implicit assumption that each of the short time segments is an individual unit with predetermined time duration. A basic problem with this assumption is that there is no well defined algorithm for determination of the duration so that the models are relevant and the assumption of stationarity is preserved. Consequently, the duration is empirically determined. A Hidden Markov Model system attempts to avoid this problem by using the same short time model for each of the steady state segments of speech samples, and a statistical characterization of how the signal changes from frame to frame. In short, an HMM does not incorporate temporal characteristics of the speech signal. The fact that Hidden Markov Models cannot effectively model temporal characteristics is a limitation of the method. Delta and

delta-delta components of the feature vector evolved partly out of attempts to include dynamic information in HMM recognition systems.

4.3.1 HIDDEN MARKOV MODEL OVERVIEW

As mentioned earlier a Hidden Markov Model is a statistical model. In order to discuss HMMs some basic concepts are provided here. Let $x_i \in \{X\}_{i=1}^{m}$ be random variables representing the model states. The random variables can be said to form a Markov chain if the probability of the current variable x_i depends only upon the previous variable x_{i-1} i.e. the following equation is satisfied:

$$P(X_i \mid X_{i-1}) = P(X_i \mid X_1, X_2, X_3, ..., X_{i-1}). \tag{4.2}$$

Equation 4.3 is a useful consequence of equation 4.2:

$$P(X_1, X_2, X_3, ..., X_{i-1}) = P(X_i \mid X_{i-1}). \tag{4.3}$$

The state sequence is directly visible to the observer in a regular Markov Model. This is not the case in a Hidden Markov Model. However, observable variables are visible and are influenced by the state variables. Each state contains a probability distribution $b_j(o_t)$ where o_t is the observation at time t and $b_j(o_t)$ is the output probability distribution at state j. Thus, the hidden parameters must be determined from the observable parameters. Figure 8 provides a visual aid to understanding these basic concepts. The HMM in the figure is a six state left-to-right HMM with observation symbols o_1 through o_6.

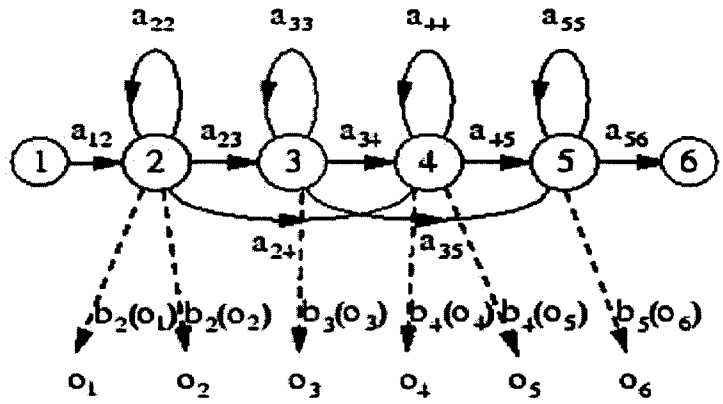

Figure 8 Six State Left-to-Right HMM.

A more concrete understanding of the Markov Model concept can be achieved by considering a short example. One person is in a closed room. Another is in a room with six jars containing colored balls. The person with the jars uses a six sided die, or some other stochastic process, to determine a jar for selecting a colored ball. Only the color of the selected ball is reported to the person in the closed room. Thus, the observation symbols would be the colors of the selected balls. To the person in room one the sequence of jars is hidden. Thus, the hidden variables would represent the sequence of jars from which the colored balls were chosen. It should be clear that there are two stochastic processes taking place here. When this happens it is described a doubly stochastic process.

Without referring to speech patterns one could view a Markov Model scenario as follows: Given a sequence of observed symbols: Á = $\{a_i\}_{i=1}^{m}$ where i is a time index and a sequence **W** = $\{w_j\}_{j=1}^{n}$ which belongs to some larger reference set **V**. The probability that **W** was the input sequence is denoted by

P(**W**|Å). The objective of a Hidden Markov Model system is to pick the most likely sequence, $\hat{W}$, given the observed symbols denoted by:

$$\hat{W} = arg\max_{w} \text{ P(W|Å)}. \tag{4.4}$$

In order to apply Markov Model Theory to word recognition we use the symbols from above to denote the system vocabulary, **V**, observed word sequences, Å = $\{a_i\}_{i=1}^{m}$, and word strings, **W** = $\{w_j\}_{j=1}^{n}$. The words in the string are elements of the vocabulary. Although Hidden Markov Model speech recognizers can model phonemes, syllables, or other sub-word units this work implements word models, specifically digits. Thus, each of the word strings, w_j in **W,** is a string of connected digits, where each digit within w_j is in the vocabulary **V**, i.e. $W \subset V$. Thus, there must be one Hidden Markov Model for each word in the vocabulary. The word models are computed so that the probability of a particular speech sample being produced given a particular word model can be determined. In other words, the word model with the highest likelihood is selected as the model of the word that was observed.

The maximization in equation 4.4 is achieved by using Bayes' Formula to express P(W|Å) as follows:

$$P(W \mid Å) = \frac{P(W)P(Å \mid W)}{P(Å)}. \tag{4.5}$$

Since the observation sequence Å is fixed it can be ignored during the maximization. Therefore equation 4.4 can be maximized by maximizing the equation,

$$\hat{W} = arg\max_{w} P(W)P(Å \mid W). \tag{4.6}$$

4.3.2 DEFINITION OF HIDDEN MARKOV MODELS

From the earlier example, the jars containing colored balls could be thought of as states and the observed colors as the possible outputs from that particular state. It should be clear that determination of the number of states can be quite difficult. As a result, in a real experiment this information is frequently an empirically determined parameter. In general a Hidden Markov Model selects and optimizes state transition probabilities, output probabilities for each state, and initial state probabilities in order to best explain an observed output sequence [58]. In order to implement Hidden Markov Models for speech recognition the following assumptions are normally made [58], [64]:

1. There are a finite number of states, N.
2. A new state is entered for each clock time, t. The state depends only on the previous state as determined by the state transition probability distribution. The transition may be from a state to itself.
3. An observation output symbol is produced after each transition. The output observation probability depends only upon the current state. Thus, there are N output observation probability distributions.

The essential framework of Hidden Markov Modeling provides a general overview but there are design problems which must be addressed before a real speech recognition system can be implemented with Hidden Markov Models. The three most fundamental implementation problems are discussed in the next section.

4.3.3 FUNDAMENTAL HMM DESIGN PROBLEMS

There are three fundamental problems that had to be solved before Markov Models could be implemented in real problems [48], [52], [54], [58]. The problems are listed below:

1. The Evaluation Problem: Given an HMM and an observation sequence compute the probability, or likelihood, of the observation sequence.
2. State Determination Problem: The determination of a best sequence of model states.
3. Training Problem: The optimization of the model parameters to best describe the occurrence of observation sequences.

The Baum-Welch algorithm is an iterative method for solving problem three and provides an efficient and elegant solution for obtaining optimal model parameters. Problem two can be solved with the Viterbi algorithm. Its solution results in knowledge of the state sequences which are required for segmenting the training sequences into states so that state occupation statistics can be used for improvement of model parameters. The forward-backward algorithm solves problem one. The probabilities computed with the forward-backward algorithm are used in performing recognition. The details of each of these algorithms can

be found in texts on Markov Chains, and "Statistical Methods for Speech Recognition," by Frederick Jelinek. This book gives a thorough introduction to Hidden Markov Modeling for speech recognition.

4.4 THE HIDDEN MARKOV MODEL TOOLKIT (HTK)

The HTK toolkit is a powerful software package which provides numerous modules, called tools, for building Hidden Markov Models [59]. It is widely used by researchers in automatic speech recognition. Although HTK is better known for its use in Hidden Markov Modeling the HTK package also has a tool for front-end analysis that is capable of computing several types of speech signal representations. Therefore, it can be used to create an entire automatic speech recognition system. The HTK front-end analysis tool, HCopy, was used to build the delta and delta-delta coefficients for the base line recognition system for this dissertation, using MFCC feature vectors computed with the WI007 front-end analysis tool.

While there is active research in the area of improving speech recognition by adapting Hidden Markov Models the main focus of this dissertation is in the extraction and shaping of speech signal information for the purposes of improving speech recognition in noisy conditions. Therefore, in order to concentrate on the front-end signal processing, the Hidden Markov Model Toolkit (HTK) was used for building word level Hidden Markov Models after our front-end produced representative feature vectors. In other words, the tools provided by the HTK package for computing HMMS was utilized but the HTK front-end analysis tool was replaced by a front-end analysis package created here at the

Old Dominion University Speech Communications Laboratory. The only exception was for the baseline (WI007) recognition system for which the HTK analysis tool was used to compute the delta and delta-delta coefficients.

The real power of the HTK package comes from the tools for HMM initialization, training, and recognition. Each tool accepts a variety of configuration parameters, allowing the user to specify a seemingly infinite number of recognizers. The core HTK tools are listed below followed by a Figure 9 which gives a flow chart for a typical training sequence.

- HCopy for signal analysis
- HCompV for HMM initialization
- HERest for iterative training
- HHed for editing of the models
- HVite performs Viterbi based recognition
- HEResults is used for performance evaluation

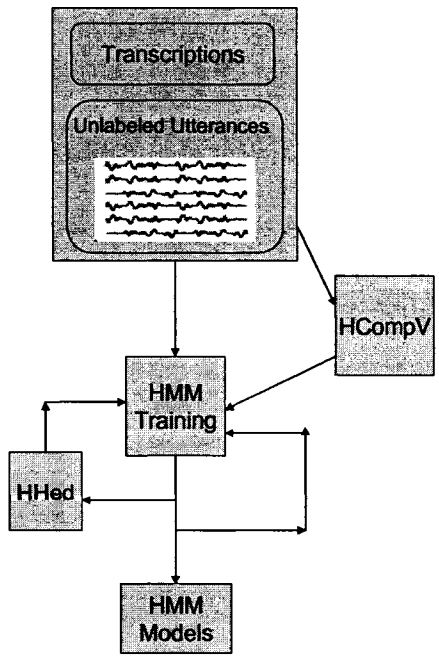

Figure 9 Hidden Markov Model training sequence.

In the iterative training sequence depicted in Figure 9 the HCompV tool accepts a set of prototype HMMs along with the parameterized speech vectors, computes the global mean and variance, and initializes the state means and variances of each model equal to the global mean and variance. The initialized models have numerous states but normally each state consists of one Gaussian mixture component. At this point the iterative stage begins with HERest. Initialized HMMs are read by HERest along with the entire set of training sentences. The data is used to accumulate the statistics for state occupation, means, and variances for each HMM. After all of the training data has been processed the accumulated statistics are used to re-estimate the HMM parameters. The HMM definition editor HHEd clones models into context-dependent sets, applies parameter tying, and increments the number of mixture components in specified state distributions. The new models are sent then back to HERest for re-estimation of the parameters. As the model complexity increases the need for more data can become a problem. Parameter tying is a method of pooling the data so that shared parameters can be better estimated.

If the iterative training procedure converges the system outputs one model for each word in the system vocabulary. These models are then used by the HVite tool for recognition. HVite requires a word dictionary, a network of allowable word sequences, and the HMMs produced by the iterative training stage. In connected digit recognition the word networks are simple word loops in which any digit can follow any other digit. However, in more general continuous speech recognition the networks are directed graphs representing a finite-state

task grammar. The dictionary defines the pronunciation for each word. HVite uses a modified Viterbi based algorithm called token passing to associate the appropriate HMM to each word instance, then performs recognition on the test data. The output is a set of transcriptions of the test set.

Finally, the transcriptions of the test set are compared with the reference transcriptions. HEResults performs this step and computes performance statistics based on the results. This is accomplished by aligning the reference and test set transcriptions and counting substitution, deletion, and insertion errors.

CHAPTER V
CONNECTED DIGIT RECOGNITION WITH LP SPECTRAL ANALYSIS

5.1 INTRODUCTION

Linear prediction has been widely used in many areas of science and engineering. Modern control problems, time series analysis, geophysics, and spectral estimation in signal processing are a small sampling of the wide applications of linear prediction. With respect to speech signal representation, parametric modeling techniques became popular in the early 1970's [81]. Currently the main use of parametric modeling for speech is for compression algorithms and speech production rather than for speech recognition. Additionally, LP derived coefficients have more commonly been computed in association with perceptual linear prediction (PLP) and filter bank speech recognizers [81]. The linear predictive analysis used in this work differs in that the final feature vectors are computed using a Discrete Cosine Transform expansion of the entire spectrum rather than the typical filter bank derived coarsely-sampled spectrum, and in the use of long window durations combined with long block lengths used for computing spectral/temporal features.

Static spectral feature components are first computed in the typical manner using single frames of the windowed signal. The length of time over which each set of static components is valid is referred to as the frame duration and the time between successive parameter computations is called the frame spacing. Additionally, at the ODU Speech Communications Laboratory the dynamic components, which are determined from the static parameters, are

computed using block processing, where a block consists of several adjacent frames. The combination window duration, frame spacing, and block length directly affect the ability of the signal model to capture spectral dynamics of the speech signal. This is particularly true in noisy environments.

In noisy environments parts of the speech spectrum are degraded by noise. For instance, in a telecommunications environment additive and convolutional noise are frequently present. Convolutional distortion is induced when the speech signal passes through telecommunications equipment such as mobile handsets and terminal equipment. Environmental noise is additive and varies depending on the location in which the speech is produced. In both cases machine recognition performance is degraded. Figure 10 illustrates noise in the spectrum of one frame taken from the digit string "75." The spectrum of the same frame is used for each of the examples depicted in Figure 10. The first graph illustrates one frame of the spectrum of the clean digit string along with the spectrum of the same frame but with a signal-to-noise ratio of 15 dB. The second example shows the spectrum of the same frame where the SNR is 10 dB and the final plot shows the spectrum of the same frame with both additive noise and convolutional distortion, at a 15 dB SNR. The spectrum becomes correspondingly more corrupted as the signal-to-noise level decreases. As can be seen in the second and third examples, the spectrum is completely overwhelmed by noise in the higher frequency regions from around 2 kHz to 3.8 kHz. Speech recognizers trained on clean speech typically exhibit poor performance when attempting to recognize sentences like those with the noise in

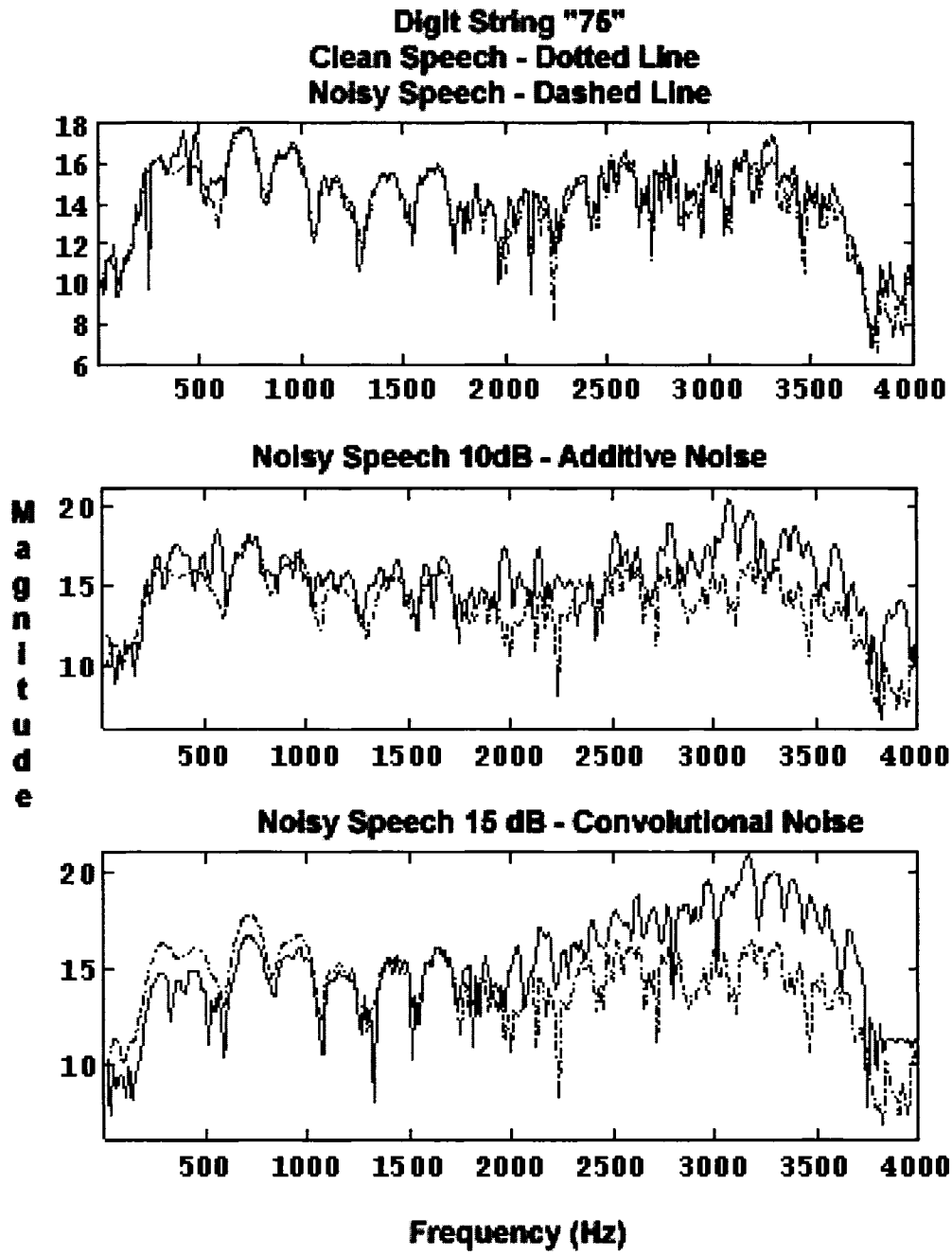

Figure 10 The spectrum of one frame of the clean digit string "75" compared with the spectrum of the same frame with additive noise at SNR 15 dB and SNR 10 dB, and with convolutional noise.

Figure 10. Even recognizers trained in similar noise to that present in the test data often do not perform with better than 93% accuracy [62], [63]. Figure 10

shows examples where large portions of the low amplitude areas of the spectrum have been lost. Thus, those regions cannot be estimated with any high degree of confidence. With this in mind we hypothesized that performance would improve if the spectral peaks were not only preserved but also emphasized or enhanced. Therefore, the goal of this remainder of this chapter is to investigate ASR performance in noise with spectral peak enhancement achieved by LP signal analysis followed by representation with DCTC and DCS coefficients.

The remainder of this chapter provides a brief overview of LP analysis for speech processing and the details for LP derived DCTC/DCS based signal modeling for the connected digit recognition task described in chapter 3. The control experiment based on the speech signal representation produced by the ETSI WI007 front-end and the baseline experiment which used the signal model determined by FFT derived DCTC/DCS features are presented in Section 5.3. An evaluation of the LP based DCTC/DCS features is reported in Section 5.4, with performance compared to those of the control experiments. Finally, a more direct comparison of the WI007 MFCC and LP derived DCTC/DCS signal models was made using varying block lengths for computing the dynamic terms of the MFCC signal model. The experiment and performance results are presented in Section 5.5. Chapter conclusions are presented in Section 5.6.

The mean square error signal analyses for computing LP based DCTC/DCS coefficients are covered in the following section.

5.2 LP SIGNAL PROCESSING

Let $s(n)$ represent the speech signal. Then the error estimate can be expressed as:

$$e(n) = s(n) + \sum_{i=1}^{N} lp(i) * s(n-i) \tag{5.1}$$

Taking the Z transform of equation 5.1 gives:

$$\begin{aligned} E(Z) &= S(Z) + \sum_{i=1}^{N} LP(i) * S(Z) * Z^{-i} \\ &= S(Z) * (1 + \sum_{i=1}^{N} LP(i) * Z^{-1}) \\ &= S(Z) * \sum_{i=0}^{N} LP(i) * Z^{-1}, \qquad LP(0) = 1 \end{aligned} \tag{5.2}$$

The second factor in equation 5.2 can be defined as the LP inverse filter:

$$H(Z) = \sum_{i=0}^{N} LP(i) * Z^{-1}, \qquad LP(0) = 1 \tag{5.3}$$

and the signal model is

$$S(z) = \frac{G}{H(z)} \tag{5.4}$$

where G is the model gain. The solution of equation 5.3 gives the LP coefficients and provides the required components for the signal model to be determined.

Mean-square error minimization of equation 5.3 leads to the matrix equation:

$$R_n(k) = \begin{bmatrix} R_x(1,1) & R_x(1,2) & . & . & . & R_x(1,N) \\ R_x(2,1) & R_x(2,2) & . & . & . & R_x(2,1) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ R_x(N,1) & R_x(N,2) & . & . & . & R_x(N,N) \end{bmatrix} * \begin{bmatrix} LP(1) \\ LP(2) \\ . \\ . \\ . \\ LP(N) \end{bmatrix}$$

where $R_x(i,k) = \frac{1}{N} \sum_{m=0}^{N-1-k} s(n+m-i)s(n+m-k).$ (5.5)

Values outside of the summation range are considered to be zero, and the assumed stationarity of the signal results in the auto covariance functions, $R_x(k)$, being functions of time differences, $t_{i+1} - t_i = \Delta t$, and the major diagonal components being equal to $R_x(0)$. Noting these properties $\overline{R}_X$ can be rewritten as:

$$\overline{R}_X = \begin{bmatrix} R_x(0) & R_x(Dt) & . & . & . & R_x(NDt) \\ R_x(Dt) & R_x(0) & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ R_x(NDt) & R_x(NDt) & . & . & . & R_x(0) \end{bmatrix}. \quad (5.6)$$

Such a matrix is said to be Toeplitz, which is a matrix in which each descending diagonal element from left to right is constant. This is beneficial because the LP coefficients can now be computed efficiently using the Levinson-Durbinson recursion [64]. Although the Levinson-Durbinson recursion method is well documented in the literature the equations are given here for completeness:

Initialization:

$$E_{LP}^{0} = R_n(0) \tag{5.7}$$

$$r_{LP}(i-1) = -\frac{R_n(i) + \sum_{j=1}^{i-1} LP^{(i-1)}(j) R_n(i-j)}{E_{LP}^{i-1}}$$

$$LP^{(i)}(i) = r_{LP}(i-1) \tag{5.8}$$

For $1 \le i \le N_{LP}$ such that $0 \le |r_{LP}(i-1)| \le 1$

$$LP^{i}(j) = LP^{(i-1)}(j) + r_{LP}(i-1) LP^{(i-1)}(i-j), \text{ where } 1 \le j \le i-1 \tag{5.9}$$

$$E_{LP}^{(i)} = (1 - r_{LP}(i-1)^2) * E_{LP}^{(i-1)}. \tag{5.10}$$

After the Levinson-Durbinson recursion the spectrum can be determined from the LP coefficients and the gain as follows:

$$S(f) = \frac{G}{\sum_{i=0}^{N} LP(i) * e^{-j2\pi(f/f_s)}} \tag{5.11}$$

$$G = \sqrt{E_{LP}^{(N)}}. \tag{5.12}$$

After computing the LP coefficients and the LP spectrum, as described above, the LP spectrum can be used in place of the FFT spectrum for subsequent processing. In particular, the DCTC features can be computed from the LP spectrum. The resultant features are referred to as LP derived DCTC coefficients. They represent the static spectral information in the speech signal and are therefore also referred to as static feature vectors. In this work the temporal (or dynamic) changes in the spectrum were computed by a 3 term discrete cosine expansion (DCS) of the static feature vectors. Thus, the dimension of the final 39 component LP derived DCTC/DCS feature vector space

is higher than that of the 13 component LP derived DCTC static feature vector space.

The experiments described in this chapter were designed to investigate noise robustness of the LP based DCTC/DCS signal representation. Evaluations were performed using the Aurora 2.0 multi-condition training data and the clean training data. The performance of a recognizer trained with multi-condition data is typically better than that of a recognizer trained with clean data. This degraded performance is presumed to be due to the recognizers' lack of knowledge of the noise characteristics in the test data. Additionally, as the signal-to-noise level decreases the recognizers' performance rapidly degrades. The mismatch in the training and test data contributes to the decrease in recognizer performance. Mismatches in training and test data are meant to simulate a scenario in which signal noise is not known a priori. Thus, improved recognizer performance with clean training only, and the resulting mismatches in the training and test data, is an important goal in ASR.

Section 5.3.1 briefly describes the signal analysis performed by the WI007 front-end algorithms developed by the ETSI working group and presents the performance, in terms of the word accuracy percentages, of the resulting automatic speech recognizer evaluated with the multi-condition and clean training data from the Aurora 2.0 database. As mentioned in Chapter 3 the WI007 front-end was designed for evaluation and comparison of front-end algorithms using the Aurora 2.0 speech data and a standardized HMM back-end which was also defined by the Aurora DSR working group [57]. In addition, the

ODU baseline DCTC/DCS features were evaluated with the Aurora 2.0 database and results compared to the Aurora control experiments presented in Section 5.3.1. An evaluation of the new LP based DCTC/DCS features is presented in Section 5.4. The performance of the new features is compared to those obtained from the control features and from the ODU baseline DCTC/DCS features. Additionally, the best parameters found in the experiments with the ODU baseline features were used as a starting point for finding the LP based DCTC/DCS signal representation which produced the best recognizer performance. The evaluations reported in Section 5.4 were also performed using the Aurora 2.0 multi-condition and clean training data. The experiments presented in section 5.5 were designed to make a more direct comparison to the WI007 MFCC signal model by varying the window lengths used to compute the dynamic coefficients from the WI007 MFCC static feature vectors. Chapter conclusions are reported in section 5.6.

5.3 CONTROL EXPERIMENTS

LP based DCTC/DCS signal processing was presented in the previous section. In this section, two control experiments are presented. The performance of the WI007 MFCC front-end has been used as a standard for the purposes of comparing evaluations of speech signal models using the Aurora 2.0 speech data. Therefore the decision was made to use the same features for our first control method. Since the signal processing for the MFCC and LP based DCTC/DCS signal models are quite different the second control experiment was performed using the FFT based DCTC/DCS signal representation, which makes

possible a reasonable comparison to the MFCC features in the first control experiment. The evaluation of the FFT based DCTC/DCS signal model also serves as a second baseline for the evaluation of the LP based DCTC/DCS features presented in section 5.4. Additionally, the optimal parameters obtained from the second control experiment were used as a starting point for the LP based DCTC/DCS features evaluated in section 5.4. Note that since the evaluation of the signal representation was the objective of each of the experiments presented in this dissertation the same Hidden Markov Model back-end was used for the recognition phase for each evaluation. The HMM configuration was described in Section 3.2.3.

5.3.1 MFCC FEATURES

The Aurora DSR Working Group designed the WI007 front-end so that a baseline speech recognition system could be realized and used for comparison of the performance of front-end algorithms developed by different research groups. Therefore, this first control experiment used the standard MFCC feature vectors as defined by the Aurora DSR Working Group in task WI007 [59]. Since the WI007 front-end only produces 14 component vectors containing static speech information the HTK signal analysis tool was used to compute time derivatives (referred to as delta and delta-delta coefficients) from the MFCC derived static features. These dynamic coefficients were computed using the regression formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2} \tag{5.13}$$

where d_t is a delta coefficient at time t computed in terms of the corresponding static coefficients $c_{t+\theta}$ to $c_{t-\theta}$. The delta coefficients are used with equation 5.13 to compute the acceleration (or delta-delta) coefficients [84]. The standardized Hidden Markov Model recognizer determined by the final 39 component MFCC feature vectors served as our primary baseline for comparison purposes.

The Aurora WI007 front-end is a cepstral analysis (MFCC) scheme for which 14 feature components per frame are computed. A notch filter performs signal offset compensation, and pre-emphasis is performed with a factor of 0.97. Subsequently, the logarithmic frame energy and C_0 coefficient are computed along with 12 Mel frequency cepstral coefficients [57]. These vectors were determined using a speech frame length of 25 ms with a 10 ms frame spacing and represent static information in the speech signal. As mentioned earlier the signal analysis tool in the HTK toolkit was used to compute the dynamic components from the static feature vectors. The zero-th cepstral term was used to compute the dynamic terms but was not included in the 13 component static feature vector. Table 10 and Table 11 give word accuracy rates for the recognizer determined by 13 MFCC terms augmented with 13 delta and 13 delta-delta terms. Word accuracy was computed using the formula:

$$\% \text{ accuracy} = \frac{\text{N-D-I-S}}{\text{N}} \tag{5.14}$$

where N is the total number of words in the reference transcription, D is the number of deletion errors, I is the number of insertion errors, and S is the number of substitution errors. Note that the results from this experiment were nearly identical to those reported by the ETSI working group, as given in Table 1 in Chapter 3. As noted in Chapter 3 for each set of results reported in this dissertation the averages are computed with the following weight function,

$$W=0.4*TSA+0.35*TSB+0.25*TSC \quad (5.15)$$

where TSA, TSB, and TSC are the averages of the percent word accuracy for each of the test sets, respectively.

Table 10 Word Accuracy for MFCC Analysis. Multi-Condition Training.

Word Accuracy (%)			
Test Set A	**Test Set B**	**Test Set C**	**Weighted Average**
89.16	87.69	85.7	87.77

Table 11 Word Accuracy for MFCC Analysis. Clean-Condition Training.

Word Accuracy (%)			
Test Set A	**Test Set B**	**Test Set C**	**Weighted Average**
69.96	66.96	71.77	69.36

5.3.2 FFT BASED DCTC/DCS SPECTRAL FEATURES

The control experiments presented in this section were designed to evaluate the frame level FFT based DCTC/DCS features computed as described in 2.1.4. Thus, performance of these features is compared to the Aurora baseline produced by the WI007 front-end recognizer, and to the performance of

LP based DCTC/DCS features computed and evaluated in section 5.4. The first experiment evaluated the signal model using the Aurora 2.0 multi-condition training data, followed by the evaluation using clean training.

For the first control experiment presented in this section the best signal model representation was determined by varying the block level processing of the frames over a range of 3 to 25 frames per block with all other parameters held fixed. Percent word accuracy for each of the block lengths are given in Figure 11. As can be seen in the figure the recognizer with a block length of 3 yielded the worst performance, indicating that 55 ms of the speech signal is not sufficient for capturing relevant speech information. It is likely that slowly varying regions of the speech signal spectrum were not well represented by this recognizer. On the other extreme, the recognizer with block length 25 performed significantly better than the one with a block length of 3, but resulted in a word accuracy approximately 4% lower than the recognizers with block lengths of 9 and 10. The performance of the recognizer with block length 25 indicates this recognizer's inability to effectively represent rapid changes of the spectral dynamics as opposed to the problem encountered with the shorter block lengths.

The best performance achieved was 83.61%, accomplished with a block length of 10. Thus, the dynamic feature components were computed over 125 ms segments of the speech signal, which is significantly longer than the typical length resulting from the more typical 25 ms window duration, 10 ms frame spacing, and 3 frames for computing dynamic features.

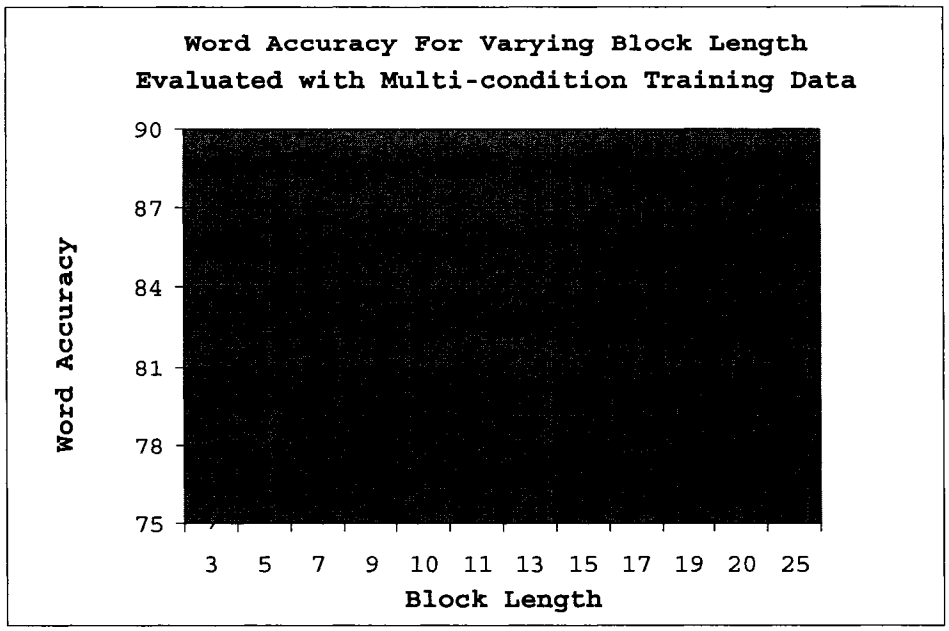

Figure 11 Word accuracy for the recognizer determined by varying the block length from 3 to 25 blocks per frame.

Figure 11 depicts the trend over the range of block lengths but does not provide details for the individual cases. Word accuracies for the best case, block length 10, are presented in Table 12, which shows that the performance for clean speech and SNR 15 are comparable to results achieved with the WI007 front-end. However, as the SNR decreases the word error rate (WER) of the FFT based DCTC/DCS recognizer increases, resulting in an average WER of 16.4% as compared to WI007 front-end word error rate of 12.2%. Thus, the baseline FFT based DCTC/DCS signal model is not as noise robust as the WI007 front-end model.

Table 12 Word Accuracy for DCSC/DCT Analysis. Block Length 10.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
Clean	98.76	98.76	98.65	98.73
SNR20	97.59	96.73	96.67	97.06
SNR 15	96.29	93.69	94.43	94.92
SNR 10	92.41	87.43	87.83	89.52
SNR 5	80.57	72.08	70.12	74.99
SNR 0	52.72	44.80	38.69	46.44
Average	86.39	82.25	81.06	83.61

The second control experiment presented in this section is the evaluation of the FFT based DCTC/DCS signal model with the Aurora 2.0 clean training data. Due to the mismatch between training and test data the performance of the recognizers were expected to degrade when evaluated with the Aurora 2.0 clean training data. Also, from Figure 11 it can be seen that the best performance for the evaluation with the multi-condition data was for block lengths in the range of 7 to 17. The performance of the recognizers trained with the clean data was not expected to exceed that of the recognizers trained with the multi-condition data. Therefore, the block length was varied within this range. All other parameters were the same as those used for the evaluation using the multi-condition training data. For block lengths 9 through 17 the word accuracies for this experiment are provided in Figure 12, which shows that the best performance was achieved with block length 13.

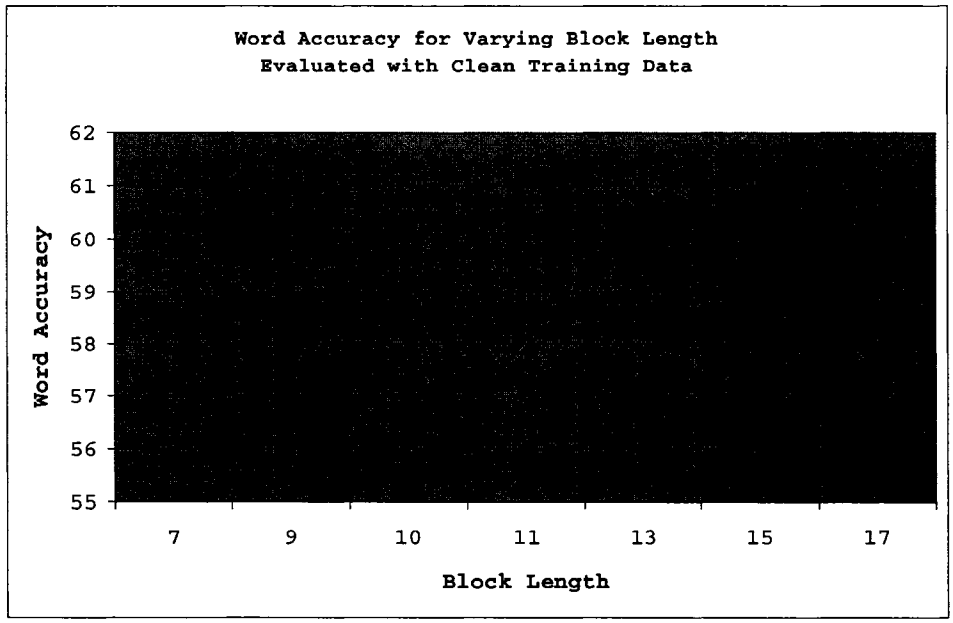

Figure 12 Word accuracy for varying the block length evaluated with clean training data.

As mentioned earlier, there is a high degree of mismatch in the training and test data. Thus, the recognizer determined by clean training data does not "learn" about any of the types of additive noise in the test data. This assumption is supported by the performance of the recognizers trained with the clean data, which achieved a best performance achieved of 61.25% word accuracy, accomplished with block length of 13. Thus, there was a relative difference of 26.74% between the performances of the recognizer determined by the multi-condition training data and the one evaluated with clean-condition training data. The best performance of the recognizer determined by the WI007 front-end trained with clean data was 66.9% word accuracy. Thus, there was a relative difference of 8.45%.

5.4 LP BASED DCTC/DCS SPECTRAL FEATURES

In the previous section two control experiments were described and results were reported. The first control experiment was conducted to reproduce the results achieved by the ETSI working group using MFCC features from the WI007 front-end analysis scheme. Performance of the WI007 recognizer was reported in [57]. In order to reproduce the ETSI published results the WI007 front-end package was obtained by the ODU Speech Communications Laboratory, and the ETSI results reported in Table 10 were reproduced to within 4 decimal points. The second control experiment used the ODU Speech Communications Laboratory FFT based DCTC/DCS signal representation method. Results from these control experiments show that there was a 4.2% difference in the word error rates achieved by the MFCC features from the WI007 front-end and the baseline FFT based DCTC/DCS features.

In this section experiments performed to evaluate the LP based DCTC/DCS features are reported. The objective of this series of experiments was to systematically test the hypothesis stated in Section 5.1, that recognizer performance would improve if the spectral peaks were not only preserved but also emphasized or enhanced. The first of the following experiments was designed to evaluate the LP based DCTC/DCS features and compare them with the performance of the control experiments. In the first experiment the numbers of Linear Prediction coefficients were varied over a range of 0 to 100, where an LP order of zero really means that LP analysis was skipped and the results are from FFT based DCTC/DCS signal analysis. A fixed block length of 10 (as

determined from the second control experiment using the multi-condition training data, described in the previous section) was used. Additionally, for the control experiments the recognizer performance was the highest for block lengths in the middle range. Therefore, for the second set of experiments with the LP based DCTC/DCS features, the block length was varied from 9 to 17 frames per block, and the LP order was again varied over the range of 0 to 100.

5.4.1 LP-DCTC/DCS SPECTRUM: FIXED BLOCK LENGTH

As previously stated the experiments presented in this section systematically tested the hypothesis that LP peak enhanced features determine a more noise robust signal model, and thus improve recognizer performance in noise. The first experiment reported in this section was an evaluation performed using the multi-condition training data. Therefore the block length was fixed to 10 (determined in Section 5.3.2) and the LP order was varied. All other parameters used for experiments reported in this section were the same as those used for the experiments presented in the previous sections. Word accuracies for the evaluation using multi-condition data are presented in Figure 13. As can be seen in the figure, the best performance was obtained with an LP order of 25 resulting in a word accuracy of 87.65%. Details for this particular experiment are presented in Table 13, with word accuracy for each test set and at individual SNR levels provided.

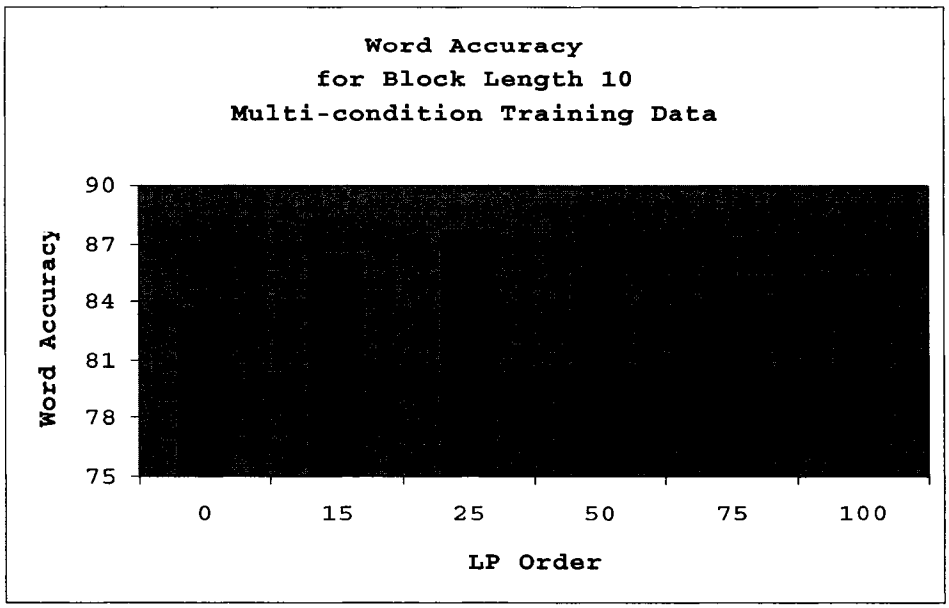

Figure 13 Word Accuracy for block length 10 and varying the number of LP coefficients, evaluated with multi-condition training data.

The recognizer determined by the LP based DCTC/DCS signal representation achieved a 12.35% word error rate, which is a significant improvement over the 16.39% WER obtained by the FFT based DCTC/DCS signal model.

Table 13 Word Accuracy achieved by recognizer determined with LP order 25 with block length 10. Multi-Condition Training.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
Clean	98.70	98.70	98.77	98.72
SNR20	97.74	96.89	97.57	97.40
SNR 15	96.97	94.74	96.41	96.05
SNR 10	94.45	90.75	93.15	92.83
SNR 5	86.16	80.20	81.88	83.00
SNR 0	63.89	56.73	50.04	57.92
Average	89.65	86.33	86.31	87.65

For the evaluation using the multi-condition training data the total improvement achieved by the LP based DCTC/DCS signal representation

resulted in recognizer performance that is essentially equivalent to that of the Aurora WI007 front-end. Additionally, the 4.83% improvement (in word accuracy) over the FFT based DCTC/DCS features is quite significant. Thus, the results of this experiment support the hypothesis that noise robustness improves with speech feature enhancement using LP based DCTC/DCS features. These results indicate that for the multi-condition training data the spectral peak enhancement achieved by the LP based DCTC/DCS signal model preserves more speech signal information. Thus, the LP DCTC/DCS features result in a signal model that is more robust to noise than the FFT based DCTC/DCS model, and is essentially as noise robust as the WI007 front-end signal representation. In the next section this hypothesis was further tested by evaluation of the LP based DCTC/DCS signal representation using varying block lengths. In addition, the evaluation of the LP based DCTC/DCS features using clean training data is presented. This evaluation is performed by varying the block length and LP order.

5.4.2 LP SPECTRUM: VARIED BLOCK LENGTH

In the previous section experiments were performed using the LP based DCTC/DCS features and recognizer performance was compared to that of the MFCC and FFT based DCTC/DCS features in the two control experiments. For multi-condition training with LP order 25 and block length 10 the performance of the new features was found to be comparable to that of the MFCC features produced by the WI007 front-end. Experiments reported in this section were performed in order to further evaluate the noise robustness of the LP based

DCTC/DCS signal model and to determine the LP order and block length that would maximize the robustness of the LP based DCTC/DCS signal features. Thus, in this set of experiments the evaluation of the LP based DCTC/DCS features with the multi-condition training data were computed with the number of LP coefficients varied from 0 to 100 and with the block length varied from 7 to 15 frames for each LP order evaluated. The 39-component LP based DCTC/DCS feature vectors were computed as they were for the results in Section 5.4.1, except for varying the block lengths, and with all other conditions identical to those reported for Section 5.4.1. Recognizer performances for each LP order are presented in Figure 14. As can be seen from the figure, the performance of the recognizer increases until the LP order reaches 25, and for this LP order results achieved with block length 11 are slightly improved over recognizer performance obtained using block length 10. Note that LP order of 25 is somewhat higher than the order typically used in methods such as PLP that do not use long block length processing.

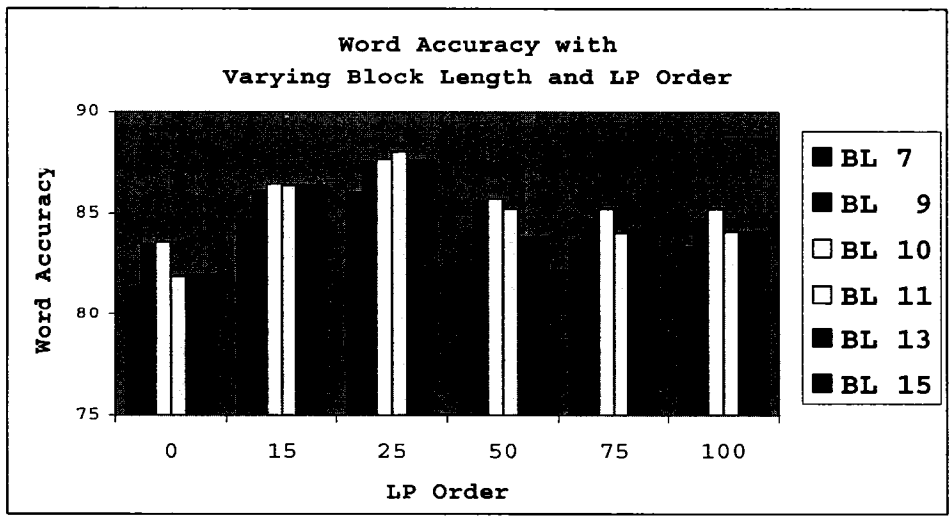

Figure 14 Word Accuracy for varying LP order and varying block length.

Thus, in these experiments, the LP order and the block length were jointly varied. The improved performance from the recognizer determined using LP order 25 and block length 11 is an indication that the longer block length combined with the peak emphasis of the model produced more robustness in the signal model. Word Accuracy for the recognizer with LP order 25 and block length 11 are given in Table 14.

Table 14 Word accuracy for recognizer determined by LP order 25 with block length 11.

SNR Level	**Word Accuracy (%)**			
	Test Set A	**Test Set B**	**Test Set C**	**Average**
Clean	98.67	98.69	98.77	98.70
SNR20	97.70	96.83	97.62	97.38
SNR 15	96.83	94.82	96.35	96.01
SNR 10	94.42	90.57	93.2	92.77
SNR 5	86.44	79.93	82.92	83.28
SNR 0	65.74	57.20	55	60.07
Average	89.97	86.34	87.28	88.03

The second experiment presented in this section is the evaluation of the LP based DCTC/DCS signal representation using the Aurora 2.0 clean training data. The control experiment using the clean data resulted in degraded recognizer performance, as compared to that of the recognizer trained with the multi-condition data. Also, the best word accuracies achieved by recognizers in the previous experiments were for block lengths 10 and 11, and the LP order which resulted in the most noise robust features was LP order 25 with block length 11. These parameters were used to determine ranges for variation of the parameters for the evaluation using the clean training data. As a result, block length was varied from 9 to 15 and for each of the block lengths the LP order was

varied from 0 to 25. Figure 15 shows the word accuracies achieved by the resulting recognizers, where the best performance was achieved with LP order 15 and block length11.

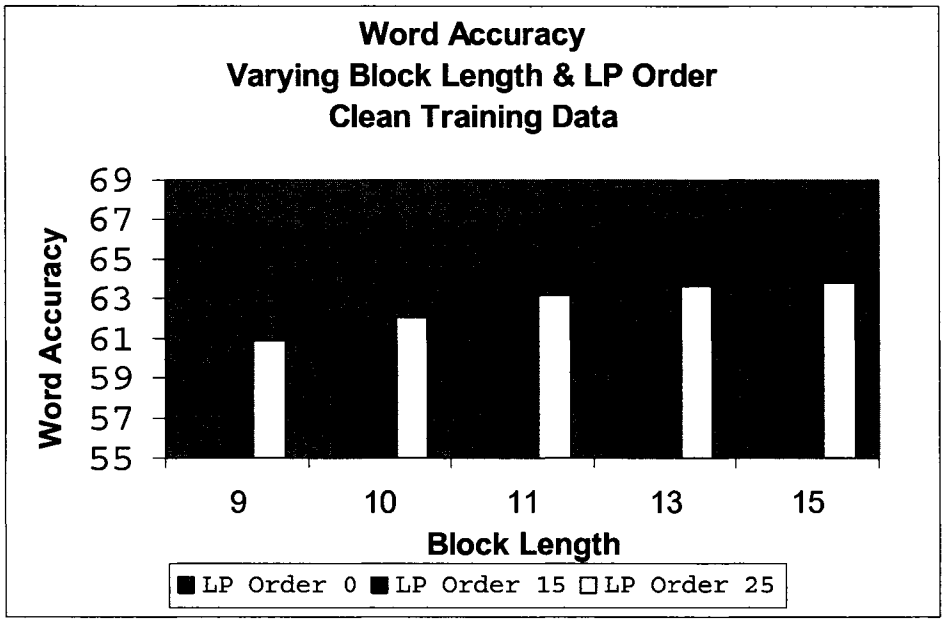

Figure 15 Word accuracy determined by varying the LP Order over 0 to 25, and varying the block length over the range from 9 to 15

The recognizer implemented with LP order 15, and block length 11, achieved 64.98% word accuracy using clean training data. Thus, there is a difference of 4.38% between the performances of the LP enhanced recognizer and the WI007 recognizer trained with the clean data.

The performance of the recognizer determined by using the improved signal model and training on the Aurora 2.0 speech data has been compared to the performance of the control experiment recognizer presented in Section 5.3.1, which was produced by using the WI007 front-end analysis scheme, also training on the Aurora 2.0 data. Thus, with respect to the multi-condition training data the

LP based DCTC/DCS features have been shown to increase noise robustness in the signal representation and to perform as well as the MFFC signal representation method presented in the first control experiment. Additionally, evaluation of the improved signal representation using the Aurora 2.0 clean data also resulted in better recognizer performance. The following section presents the evaluation of the MFCC signal model with varying block lengths, and the performance of the resulting recognizer is compared with that of the LP based DCTC/DCS signal model.

5.5 MFCC SIGNAL REPRESENTATION: VARIED BLOCK LENGTH

As described in Section 5.3.1 the MFCC analysis scheme produces static feature components only. Therefore, determination of the dynamic coefficients was accomplished using the HTK toolkit and the static coefficients from the MFCC analysis. Recall from Section 5.3.1 that the window length used to compute the MFCC coefficients was 25 ms and the frame rate was 10 ms. As defined by the standard HMM configuration for the WI007 front-end, the window length used to determine the delta terms was 3, and the window length used to determine the delta-delta terms was 2. These parameters result in what is roughly equivalent to the ODU Speech Communications Laboratory block length processing using a block length of 3 with a 25 ms window and a frame rate of 10 ms, or what amounts to 45 ms segments of the signal being used for computation of each set of dynamic coefficients. Note that in each of the other experiments presented in this dissertation the recognizers best performances

with window length of 35 ms and block lengths of either 10 or 11. As mentioned earlier, this equates to the computation of each set of dynamic coefficients over 125 ms and 135 ms, which is one of the unique aspects of the signal processing by the block method used at ODU. The WI007 front-end was designed to compute only the static coefficients with the expectation that the dynamic terms would be computed by the signal analysis tool in the HTK toolkit. Additionally, the HTK parameters were expected to be fixed. Thus, the performance of the MFCC features were expected to degrade if the dynamic information was encoded using longer windows, as compared to the block lengths used in the signal representation methods presented in this dissertation. Therefore, a set of experiments was performed in order to confirm the hypotheses regarding the MFCC front-end analysis scheme, as well as to make a more direct comparison between the two signal model methods.

In order to make the comparison discussed in the previous paragraph the experiments presented in this section were performed using the WI007 front-end algorithm to compute the MFCC static feature vectors. Note that the parameters were unchanged from the parameters used for the MFCC control experiment presented in Section 5.3.1. The HTK toolkit was then used to compute the dynamic coefficients from the MFCC static feature vectors. The only parameter varied in this experiment was the block lengths for the determination of the dynamic coefficients, which were varied over the range 3 to 11 frames.

The recognizer determined by the WI007 signal model and trained with the multi-condition data behaved as expected. Performance degraded as the

window lengths were increased, and calculation of the dynamic coefficients with window length 11 resulted in the MFCC recognizer with the worst performance, which was 16.58% WER. This is a 27.74% relative decrease as compared to the best performance of 11.98% WER achieved with the LP based DCTC/DCS signal model. Figure 16 shows the WI007 recognizer performance degrading as the window lengths are increased.

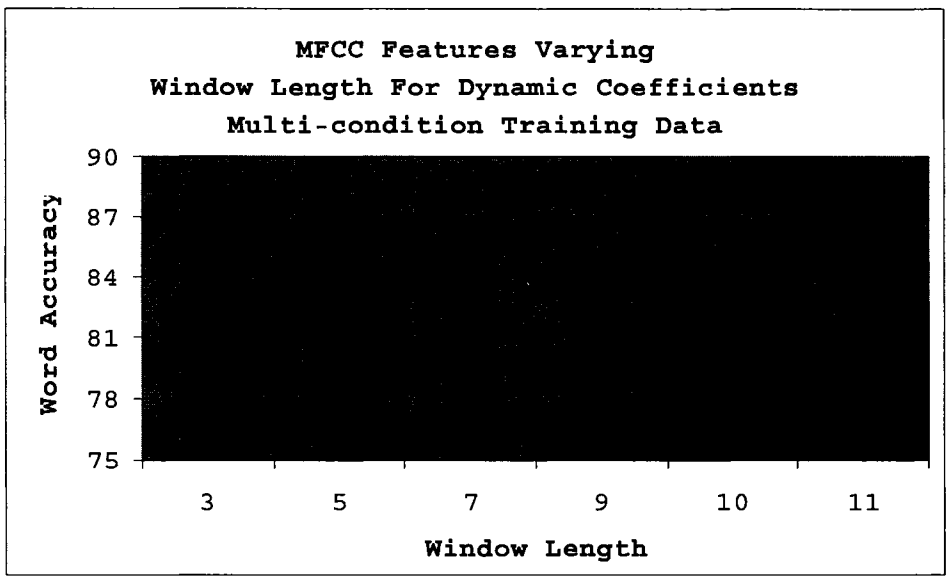

Figure 16 WI007 front-end with varying window length for computation of the dynamic coefficients.

The MFFC recognizer was also evaluated using the Aurora 2.0 clean data, with all other parameters were held fixed from the evaluation using the multi-condition data. In this case the performances achieved by the recognizers decreased until the block length reached 7 and then began to increase. A 29.05% WER was the best performance achieved (BL=10). This is a relative improvement of 12.0% WER as compared to the best performance of 33.01% WER achieved by the MFCC control recognizer. Figure 17 shows the

performances of the recognizers determined by the WI007 signal model, with varying block length used in the computation of the dynamic coefficients.

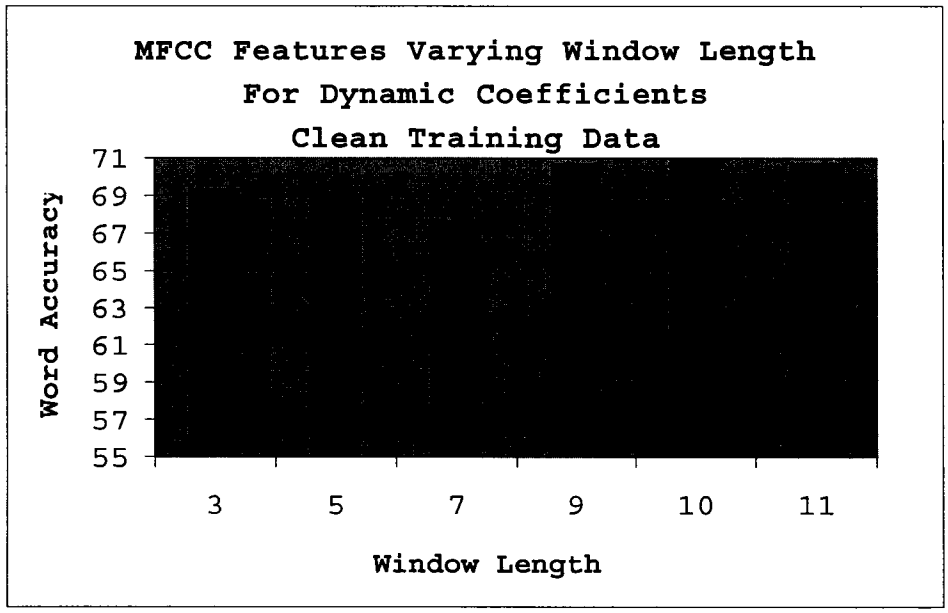

Figure 17 WI007 front-end with varying block length for computation of the dynamic coefficients.

By varying the block length used to determine the dynamic coefficients the performance of the WI007 recognizer evaluated with clean-condition training data improved while the same approach resulted in decreased performance when evaluating with the multi-condition data. Without further analysis this behavior cannot be generalized. However, performance indicates that the signal model was optimized for multi-condition training with fixed block length of 3.

5.6 CHAPTER CONCLUSIONS

The LP based DCTC/DCS signal representation with 39 terms and the standard HMM recognizer results in recognizer performance of 11.97% WER for the Aurora 2.0 database, as compared to the 12.23% WER achieved by the

WI007 front-end representation, and the 16.39% WER of the FFT based DCTC/DCS signal model. This represents a relative improvement of 2.13% WER over the Aurora WI008 front-end features, and a relative improvement of 26.97% WER over our baseline features for the same database. Noise robustness was improved due to the signal representation obtained from the longer block length processing combined with the spectral enhancement of speech signal peaks determined from the LP based DCTC/DCT analysis. Thus, the LP based DCTC/DCS signal representation has been shown to be as robust to noise as the WI007 signal representation.

The primary contribution of the work reported in this chapter was the demonstration that when the LP order is suitably chosen the magnitude frequency response estimates the envelope of the signal spectrum, and the all pole nature of linear predictive coding produces a spectral envelope with sharp peaks that, when combined with long block length processing produces a signal representation which is more robust to noise. Thus, LP based DCTC/DCS features can capture speech information in spectral peaks resulting in improved performance in noise and channel distortion. The advantage of this method is that preservation of important speech information by emphasizing spectral peaks and ignoring valleys is accomplished with essentially no additional computational demand. Additionally, the LP based DCTC/DCT features are straightforward to implement, and are slightly more robust to noise and channel distortion than the more standard MFCC coefficients with additional dynamic coefficients.

CHAPTER VI
MORPHOLOGICAL FILTERING IN THE SPECTRAL DOMAIN

6.1 INTRODUCTION

Nonlinear filtering techniques have become increasingly important in signal processing, and are often better than linear filters at removing noise without distorting signal characteristics. Historically, homomorphic and polynomial filters have been the primary class of linear filters used by the signal processing community. Recently, order statistic and morphological filters have been receiving more attention but still fall into a second, less utilized class. Homomorphic filters were developed during the 1970's and obey a generalization of the superposition principle [40]. While the design and analysis of homomorphic and polynomial filters are somewhat similar to those used for linear filters, order statistic and morphological filter design cannot be achieved using generalizations of linear techniques. Thus, these types of filters are frequently designed using heuristic methods. Consequently, the behavior of order statistic and related filters, such as morphological filters, were not well understood until important results on their statistical behavior were defined.

In the early 1980's root signals, the class of signals invariant to median filtering, were defined, and important results on the statistical behavior of the median filter presented [76]. The median filter is an order statistic filter that replaces the center value in the filter window with the median window value. A median filter is recursive if the values in the window are updated as the filter acts on the signal.

In general, design and analysis of nonlinear filters is more difficult than the design and analysis of linear filters. In 1987 statistical and deterministic analyses for the basic morphological filters were published [74]. This publication was an important contribution to signal processing since it established a firm mathematical foundation for the use of these filters. Mathematical morphology is a method of nonlinear filtering limited to maximum and minimum operations which are effective at noise suppression, and are more tractable than other nonlinear filtering techniques. In many applications a primary drawback of morphological filtering is the introduction of a deterministic bias into the filtered signal. However, for the application intended in this dissertation this bias is not a drawback since the reduction of signal valleys and enhancement of signal peaks is a specific goal.

As mentioned above an important property of a nonlinear filter is its root signal set, also referred to as fixed points of the filter. The root signal set is the set of signals that are unchanged by the operation of the filter. A root signal consists only of constant neighborhoods and edges, where a constant neighborhood is an area of constant value with length at least half the length of the window and an edge is a monotonic region of any length between two constant neighborhoods [77]. A critical goal of morphological filtering is the preservation of slowly varying regions and rapid transitions, as well as the smoothing of impulses and rapid oscillations. Thus, root signal sets with these properties are clearly quite important.

The maximum and minimum functions that define the basic morphological filtering are the cause of the signal bias mentioned earlier. For example, the close and close-open operations begin with the dilation operation, which replaces the center value in the filter window with the maximum value within the filter window. Signals filtered with these operations lie on or above the root signal found by median filtering. Filtering with the open and open-close operations result in an output signal which has a magnitude below or equal to the root signal found by median filtering. In many applications (e.g. image processing) the bias is undesirable but the smoothing property of the morphological filtering is valuable. In this dissertation the goal is to broaden spectral peaks and reduce the depth of noisy low level spectral valleys. Therefore the bias in the filtered signal can be taken advantage of with respect to the specific goal of the morphological filtering used in this dissertation. However, the amount of spectral peak broadening must be controlled so as to avoid over smoothing of the filtered spectra.

The remainder of this chapter is organized as follows. Section 6.2 presents definitions for mathematical morphology and discusses general properties of morphological filtering. Signal processing for morphological filtering is presented in Section 6.3. Experimental results from evaluations of morphologically filtered FFT based DCTC/DCS features are presented in Section 6.4. Evaluations of the morphologically filtered LP based DCTC/DCS signal model are presented in Section 6.5. The evaluations presented in Sections 6.4 and 6.5 were performed with the dilation operator. In order to demonstrate the

behavior of the other morphological filter operations an evaluation was performed for each of the signal models determined by filtering with the erosion, open, close, open-close, and close-open filters. The results of these experiments are presented in Section 6.6. Finally, the complexity of the Hidden Markov Model recognizer was increased and tested with the signal model determined by morphologically filtering the spectra of the LP based DCTC static coefficients with the close-open operator. Results of these experiments are presented in Section 6.7. The last set of experiments was an evaluation of morphological filtering and the LP based DCTC/DCS signal model using the Aurora 3.0 database. Recognizer performances are reported in Section 6.9. Chapter conclusions are presented in Section 6.10.

6.2 MATHEMATICAL MORPHOLOGY

In morphological filtering geometrical properties of the signal are modified by morphological convolution of the signal with a structuring element, which is chosen to enhance specific characteristics of the filtered signal. Variation of the shape and size of the structuring element provides a means of extracting different types of information from the signal. Thus, signal shaping is accomplished by the use of a smooth shaped structuring element.

Every morphological filter must be translation and scale invariant, depend only on local signal values, and be upper semi-continuous [74]. A morphological convolution is upper semi-continuous if the structuring element has a compact region of support. This requirement is not a problem in signal processing because every sampled signal is upper semi-continuous. There are four basic

morphological filtering operations which are derived from Minkowski set addition, defined below.

Minkowski set addition and subtraction, respectively, are given by equations 6.1 and 6.2.

$$A \oplus B = \{a + b : a \in A, b \in B\} = \bigcup_{b \in B} A_b \text{ , where } A_b = \{a + b : a \in A\} \quad (6.1)$$

$$A \square B = \{a + b : a \in A, b \in B\} = \bigcap_{b \in B} A_b \neq \varnothing \text{, where } A_b = \{a + b : a + b \in A\} \quad (6.2)$$

Equations 6.1 and 6.2 mean that the union of the translates of B produce dilation whereas the intersection of the translates of B produce erosion. The translates of B relative to A include every case of B shifted such that for each element $a \in A$ the center of B is coincident with a. Thus, the dilation of *A by the set B* is the union of the translates of B which form a non-empty intersection with A. The erosion of the set *A by set B* is the union of the set of points to which the reflection of *B* may be translated while still being completely contained within the original set *A*. All filters designed from these basic operations behave similarly to the well-known median filters [74]. Combinations of the two basic operations, dilation and erosion are used to produce *the* open, close, open-close, and close-open operations. The first stage of the open and open-close operators is erosion. Whereas dilation is the first stage of the close and close-open filters. Additionally, the geometric structure of mathematical morphology can be used to create filters that use signal threshold values to produce further improvements in noise suppression and peak detection.

If we let $B^S = \{-b : b \in B\}$ be the symmetric set of B with respect to the origin, then the operations of erosion, dilation, opening, and closing, are derived from Minkowski set addition as defined below:

$$X \oplus B^S = \{z : B_z \bigcap X \neq \varnothing\} = \bigcup_{b \in B} X_{-b} \tag{6.3}$$

$$X \square B^S = \{z : B_z \subseteq X\} = \bigcap_{b \in B} X_{-b} \tag{6.4}$$

$$X_B = (A^c \square B^S) \oplus B \tag{6.5}$$

$$X^B = (A^c \oplus B^S) \square B \tag{6.6}$$

The graphic example in Figure 18 illustrates these abstract set theoretic definitions.

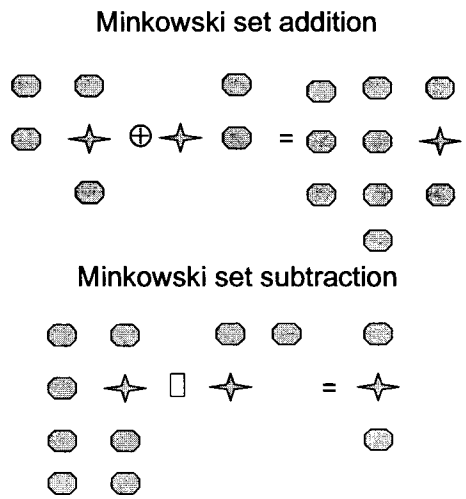

Figure 18 Example of Minkowski set addition and Minkowski set subtraction.

From the definitions above it can be seen that morphological filtering was originally derived from set theory, and therefore was applied to binary signals. Thus, the first morphological filters were set processing filters. However, the

foundation of mathematical morphology has since been extended to multi-level signals [75] [78]-[80], which led to function processing (FP) and function set processing (FSP) operations. The extension from set processing to function processing is accomplished by a process referred to as threshold decomposition of the function and accomplished by expressing the function as an ordered set of functions [75]. This is performed by taking a cross section of the function at each value in the range, and then taking the union of these cross sections over the range of the function. This process gives a complete representation of the function, i.e. the function can be recovered from the union of the cross sections. The use of signal threshold values is one way to move from set processing to function processing. However, in this dissertation an equivalent representation called the umbra of a function [78] is used.

The umbra of a closed function *f*, denoted by U(*f*), is the Minkowski sum of the graph of the function and the half-open set $(-\infty,0]$. Figure 19 shows a function g in R^2 and its umbra. One way to determine the umbra of *f* is by projecting the points in *f* onto $(-\infty,0]$. Thus, the umbra of a function can be viewed as its shadow. The umbra of a function *f* is a closed set and is analytically defined by.

$$U(f) = \{(x,t) \in DxF : f(x) \geq t\} \tag{6.7}$$

where D and F are the domain and range of *f*, respectively.

For each x in the domain of *f*,

$$f(x) = \sup_{t}\left((x,t) \in U(f) \right) \tag{6.8}$$

Thus, *f* can be uniquely reconstructed from its umbra, U(*f*).

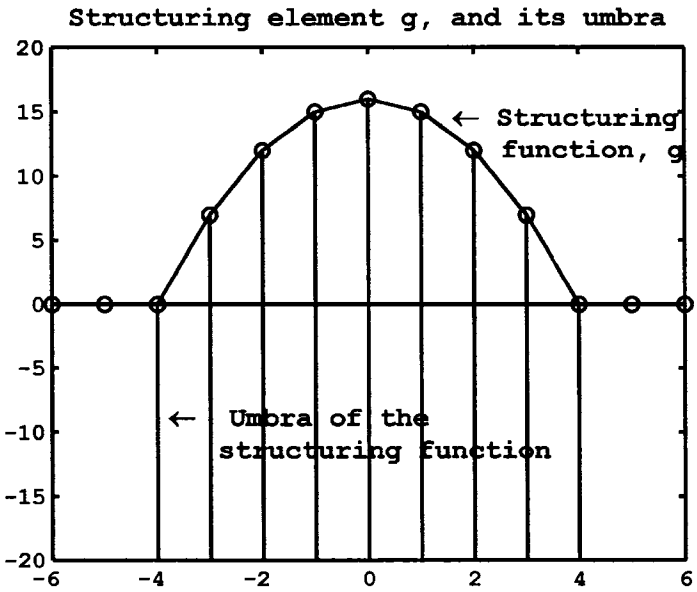

Figure 19 The graph of the structuring function g, and its umbra. The function g is the parabola in the range [-4,4] and is zero elsewhere. The umbra of g is indicated by the solid lines extending downward in the range [-4,4].

The use of the definition of the umbra of a function is equivalent to the signal threshold method and allows morphological function processing filters to be treated as special cases of set processing filters in that the function processing filters process the umbrae of the input functions instead of the set processing performed by SP filters [78]. In this dissertation morphological filtering of a signal is accomplished via the transformation of its umbra by a compact structuring function B. Using the umbrae of f and B the Minkowski addition and subtraction of f with B is defined as follows:

$$U(f \oplus B) = U(f) \oplus B = U(f) \oplus U(B) \tag{6.9}$$

$$U(f \boxminus B) = U(f) \boxminus B^r = U(f) \boxminus [U(B)]^r \tag{6.10}$$

where $B^r = \{(x,-t) \in DxF : (x,t) \in B\}$ is the reflected set of B with respect to D. From equations 6.9 and 6.10 it can be seen that the transformation of the umbra of f by B is equivalent to the transformation of the umbra of f by the umbra of B.

Set structuring elements assume a flat intensity profile over their region of support. In contrast, structuring functions process signals based on a shape based profile. Therefore, in this dissertation the umbra of the speech signals are transformed, or filtered, by the umbra of a compact structuring function, g. In order to emphasize the signal peaks and suppress low level noise the structuring function was chosen to be a parabola, defined by the following equation:

$$g = H * (N^2 - n^2) \tag{6.11}$$

The curvature of the parabola is controlled by the parameter H, and the region of support of the parabola is defined by N, which is determined by a specified frequency range. A large value for H results in a parabola that more deeply penetrates the peaks (valleys) of the function.

Figures 20 and 21 give examples of one frame of the spectrum of the digit string "008" before and after dilation and erosion with the structuring function g, respectively, where h = 2, N = 9, and the region of support is $[-4,4]$. The number of points in the region of support is computed using the formula:

$$[N] = F_{sp} * \frac{L}{F_s}, \tag{6.12}$$

where F_s is the sampling rate, L is the length of the windowed signal, and F_{sp} is the specified filter window size in Hz. For Figure 10 the region of support was

computed using the values $F_s = 8000\ Hz$, L=512, F_{sp}=140 Hz. From Figure 20 it can be seen that the dilation of the spectrum of the function by the structuring function broadens peaks and reduces the depth of the valleys of the input spectrum. Erosion of the same frame of the spectrum by g is illustrated in Figure 21. The figure shows the erosion of peaks and tracking of valleys of the input function.

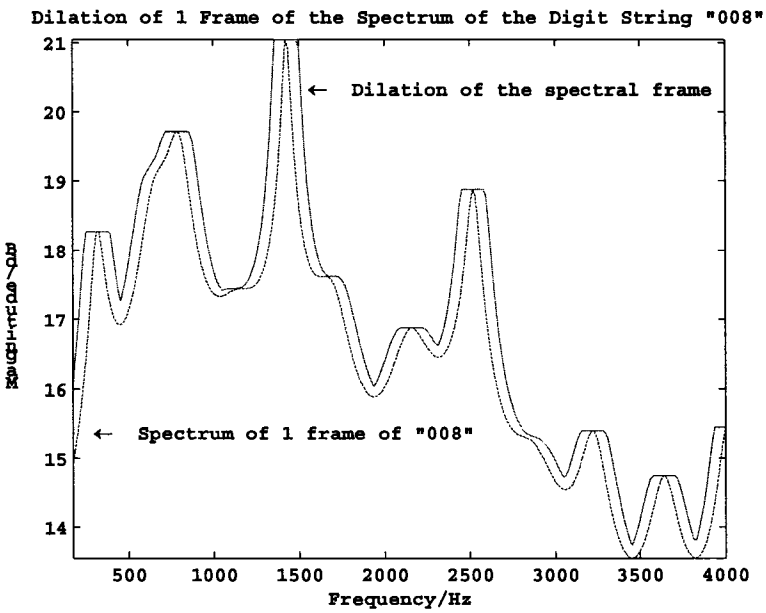

Figure 20 Dilation of one frame of the spectrum with g.

An example of the same spectral frame after the open-close by g is given in Figure 21. The graphic shows the clipping of the peaks of the input spectrum. It can also be seen that the depth of the valleys is not reduced by the cascaded operation. The structuring function was also used to perform the open-close operation on the same spectral frame.

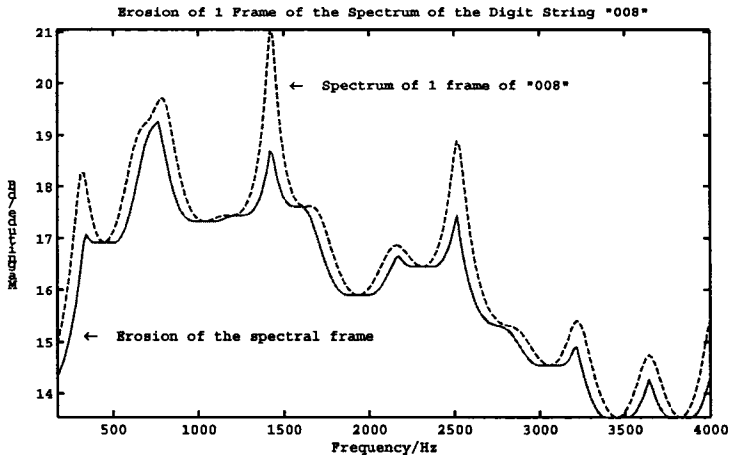

Figure 21 One frame of the spectrum of digit string "008" eroded with g.

Figure 22 depicts the same spectral frame before and after the close-open by g.

The center of the structuring function is translated to each point of the signal. As a result, the filtered signal peaks (valleys) are located at the center of the filter window. Additionally, the width of the candidate peaks and valleys is directly dependent on the size of the region of support of the structuring function g. For example, large spikes of short duration present in the signal are referred to as impulsive noise. If the width of the noise spikes does not exceed the width of the region of support of the structuring function the noisy peak (valley) can be effectively suppressed by the open-close operation, which begins with the erosion operation depicted in Figure 21. Suppression of sharp peaks can be seen in Figure 22 where peaks are reduced via the open-close operation. The sharper peaks are clearly more affected by the open-close operation.

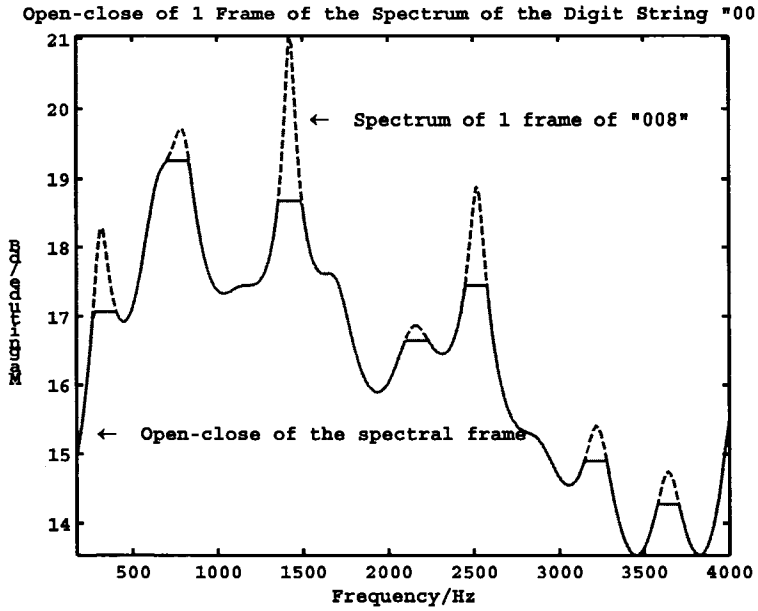

Figure 22 One frame of the spectrum of the digit string "008" after open-close by g.

Peak broadening and valley depth reduction by the close-open operation is depicted in Figure 23. Recall that the region of support of the structuring function was $[-4,4]$ for each of the Figures 19 through 22, which means the filter window length was approximately 140 Hz. Thus, spectral peak broadening produced by this window length is quite large. As can be seen in Figure 23 the filtered signal is over smoothed and is likely to result in degraded recognizer performance.

In summary, Figures 20 through 23 provide an illustration of the effects of the filters, which are quite different from each other in that erosion removes positive spikes and preserves valleys, while dilation essentially tracks the valleys and broadens peaks. Both operations preserve monotonic ranges within the signal. The open-close and close-open operations are cascaded operations built

from the basic operations of erosion and dilation, which essentially double the effective window length.

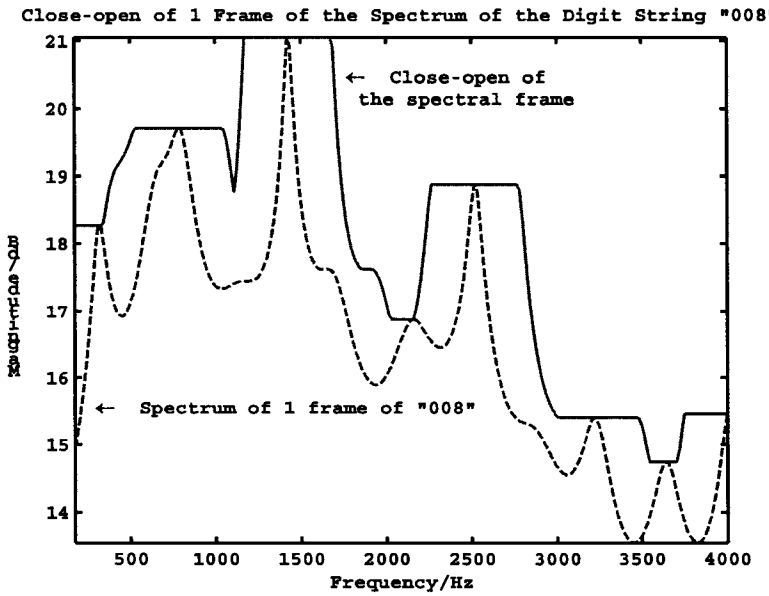

Figure 23 One frame of the spectrum of the digit string "008" after close-open by g.

The remainder of this chapter reports the results of the investigation of the use of nonlinear filters based on mathematical morphology discussed above to determine peak enhanced speech signal representations. Specifically, filters based on mathematical morphology are developed, analyzed, and applied in a variety of speech recognition experiments. Figures 20 through 23 show the basic behavior of four morphological filters, and depict the deterministic bias in the filtered signal. Morphological filtering presented in this work was designed to take advantage of this characteristic to enhance speech signal spectral envelopes. The morphologically filtered LP derived DCTC/DCS signal representation is compared, via experimental results, with MFCC, DCTC/DCS, and LP based DCTC/DCS signal representations.

6.3 MORPHOLOGICAL FILTER SIGNAL PROCESSING

Six different morphological filter operators were defined in Section 6.2. As seen in Figure 21 the erosion and open operators reduced the spectral peaks. Therefore, they were not considered useful for speech signal enhancement for ASR. Additionally, each of the operators presented in this dissertation was implemented with the structuring function defined by equation 6.11, given in the previous section. The region of support was determined by N, as specified in equation 6.12. Each of the operators open, close, open-close, and close-open are computed by cascading the dilation and erosion operators. Therefore, only the signal processing for the dilation and erosion operations are presented in this section.

Let the input signal $s(m)$ have length M. Then for each m such that $1 \le m \le M$ the filter window is defined as $W = [m-N, m+N]$. Also, the beginning and ending of each frame of the signal is padded with the first and last value, respectively such that $m \pm N \ge 0$.

DILATION

The structuring function is translated such that the center of the structuring function is coincident with the current signal value $s(m)$, which is then replaced as follows:

$$\tilde{s}(m) = \sup_{W}(s \cup g), \text{ where W is the current filter window.} \qquad (6.13)$$

EROSION

The erosion operator is defined as the intersection of the *f* and the translates of g which are properly contained in *f*. Thus, g is translated in the same manner as for the dilation operator. However, if the translate is not properly contained in *f*, it is contained in the umbra of *f*, $U(f)$. This can be seen by sliding the structuring function down along the "shadow" of *f* until $g \subset U(f)$. Subsequently, the erosion is determined by taking the minimum of the intersection over the filter window as follows:

$$\tilde{s}(m) = \inf_{W}\left(s \cap g\right), \text{ where W is the current filter window} \qquad (6.14)$$

Sections 6.4 through 6.8 present evaluations of morphologically filtered FFT and LP based DCTC/DCS signal models using the Aurora 2.0 and Aurora 3.0 databases.

6.4 FFT BASED DCTC/DCS FEATURES:MORPHOLOGICAL SMOOTHING

Experiments presented in this chapter were conducted in order to test the hypothesis that preservation and enhancement of spectral peaks, by morphological filtering of the LP based DCTC/DCS signal representation, would improve the performance of speech recognition in noise. Part of this hypothesis was tested via evaluations of the peak enhancing capability of the LP based DCTC/DCS signal representation, which were presented in Chapter 5. Specifically, Sections 5.3.1 and 5.3.2 presented control experiments which used the WI007 MFCC feature vectors and the ODU FFT based DCTC/DCS feature

vectors computed without the use of LP analysis. The LP based DCTC/DCS signal representation was evaluated and compared to the performance of recognizers from each of the control experiments.

Speech recognizers determined by the FFT based DCTC/DCS feature vectors achieved a best performance of 16.39% WER while the recognizers determined by the MFCC feature vectors performed with a word accuracy of 12.23% WER. Finally, Section 5.3.4 presented evaluations of the LP based DCTC/DCS signal representation and compared performance to that of the control experiment recognizers. The best performance achieved by the LP based DCTC/DCS features was 11.98% WER, which is a slight improvement over the performance achieved by the control features from the MFCC recognizer in Section 5.3.1.

In this section experiments are presented in which the spectrum of the LP based DCTC static feature vectors is morphologically filtered using the dilation operation with filter length In the range 0 to 400 Hz. Figure 20 gave an illustration of the behavior of the dilation of one frame of the spectrum of the digit string "008" with a window width of 140 Hz. As can be seen there the peaks of the spectrum are broadened while the valleys are left essentially unchanged. This enhancement of the spectral peaks allows more speech information to be captured and was expected to improve the noise robustness of the recognizers.

Morphological filtering with long windows can result in over smoothing of the peak regions potentially resulting in loss of detail in the spectral representation. This effect is not desirable because it can lead to degraded

recognizer performance. This theory was tested by also filtering with a window width of 400 Hz, which is more than twice as long as the 234 Hz window. Thus, it was expected that performance would be degraded by morphological filtering with such a long window. Finally, with the exception of the morphological filtering (dilation) of the spectrum of the static feature vectors, the FFT based DCTC/DCS feature vectors used in the experiments presented in this section are the same as those used in the experiments presented in Chapter 5. Recall that a window width of 0 Hz defaults to the FFT based DCTC/DCS feature vectors with no morphological smoothing. Pilot experiments were run in which the experimental setup was similar to the second control experiment in that the block length was varied from 3 to 25. These experiments were conducted in order to determine the best block length to use with morphological filtering of the spectra of the static feature vectors. Block lengths 11 and 13 were determined to result in the best performance from the pilot experiments. Therefore, evaluations presented in this chapter were all implemented with fixed block length of either 11 (multi-condition training) or 13 (clean training).

6.4.1 MORPHOLOGICALLY FILTERED FFT-DCTC/DCS: AURORA 2.0 MULTI-CONDITION TRAINING

Evaluation of signal models, determined by morphologically filtered spectra of the FFT based DCTC/DCS static components, are presented in this section. Filtering was performed with the dilation operator varied over the range [0,400] Hz. Recognizer performance from the evaluation using multi-condition training is presented in Figure 24.

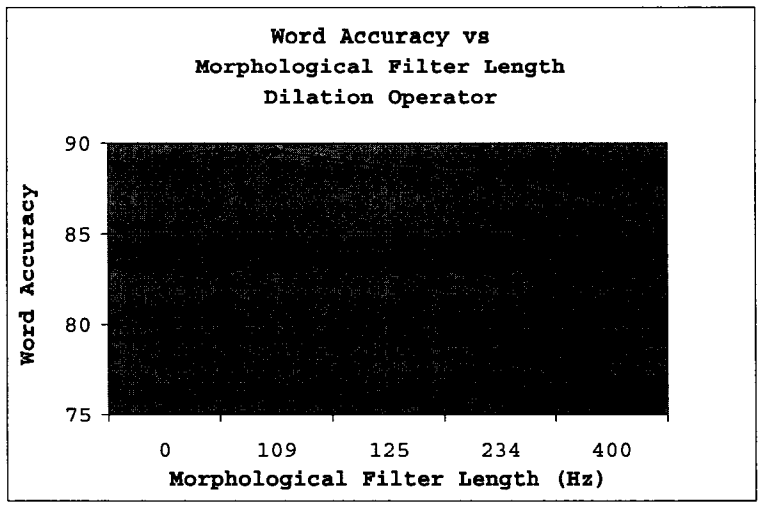

Figure 24 Word Accuracy as a function of filter length (dilation operator). The recognizer was trained with multi-condition data.

As with the experiments presented in Chapter 5 the recognizer accuracy was highest with block lengths 10 and 11. However, the addition of morphological filtering resulted in more noise robust recognizer performance. The best results were achieved by the recognizer determined by morphological filter window width of 109 Hz and block length 11. Figure 24 also shows that as predicted recognizer performance decreases as the filter length is increased. Recognizer performance for the each of the test sets are provided in Table 15 so that a comparison can be made with the performance of the recognizer built using the base line FFT based DCTC/DCS features determined in experiments presented in Chapter 5.

Table 15 Word Accuracy obtained with (dilation) filter length 109 Hz and block length 11. Multi-Condition Training.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
Average	90.76	88.61	89.02	89.57

The new recognizer achieved a word error rate of 10.43%. Whereas the recognizer determined by the FFT based DCTC/DCS signal model (presented in the second control experiment in Section 5.3.2) had best performance of 16.39% WER. Comparison of these results with those shown in Table 15 shows a reduction in word error rate of 36%. Additionally, recall from Table 12 that 11.07% WER was the best performance achieved by recognizers presented in Chapter 5. The word error rates for test sets A, B, and C were 10.03%, 13.66%, and 12.72%, respectively. Thus, the reduction in word error rates in test sets B (medium mismatch) and C (high mismatch) were 16.62% and 13.68% WER. These improvements are an indication that the new signal model is more robust to the mismatch in the training and test data. Improvements were achieved with the addition of the new morphological filtering which resulted in the preservation of more spectral peak information encoded in the static feature components, and reduction of the depth of the spectral valleys which eliminated some of the low level noise.

The relative improvement of the newly determined recognizer over the performance of the recognizer determined with the FFT based DCTC/DCS features is quite impressive. However, a more important comparison is the relative reduction in WER of 14.72% over the performance of the standard MFCC based recognizer presented in the first control experiment in Section 5.3.1, which performed with a 12.23% WER. Thus, the new recognizer is more robust to noise than the recognizer determined with the standard MFCC feature vectors.

Morphological filtering was performed before the dynamic components were computed, meaning that the spectral enhancements due to morphological filtering were encoded into the dynamic feature components. Thus the experimental results presented in this section support the hypothesis that enhancement of the spectral peaks via morphological filtering of the speech spectrum can improve the performance of automatic speech recognition in noise.

6.4.2 MORPHOLOGICAL FILTERED FFT- DCTC/DCS: AURORA 2.0 CLEAN TRAINING

In order to test the new signal model with the Aurora 2.0 clean training data the evaluation presented in Section 6.4.1 was repeated. However, preliminary experiments were performed to determine the best block length (BL=13) for clean-condition training. With the exception of block length the evaluation was performed with the same parameters used in the experiments presented in the previous section. The recognizer determined by a 125 Hz filter length obtained a 29.94% WER, which was the best recognizer performance achieved with this experimental setup. Figure 25 shows the recognition word accuracies as a function of filter length.

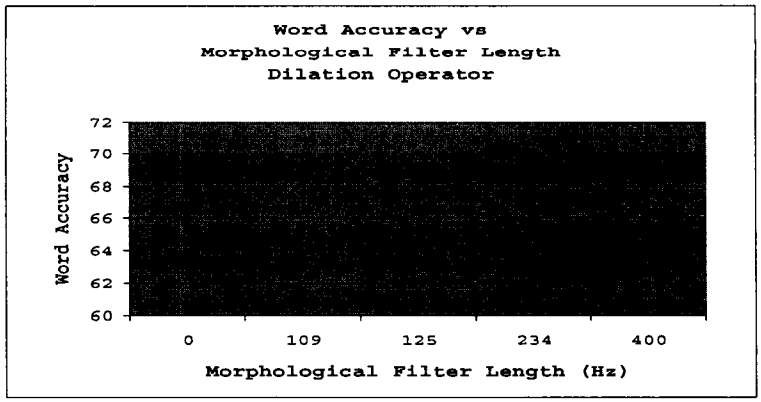

Figure 25 Performance of recognizers determined by varying the morphological filter length (BL=13), and evaluated with clean-condition training.

There was a 2.28% relative improvement in word error rate over the performance of the recognizer determined by the WI007 signal model, presented in Section 5.2.1, which achieved a 30.64% WER. Table 16 gives averages for the recognizer determined with morphological filter length 125 Hz

Table 16 Word Accuracy obtained with (dilation) filter length 125 Hz and block length 13. Clean-Condition Training.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
Average	71.35	66.39	73.14	70.06

6.5 LP BASED DCTC/DCS: MORPHOLOGICAL FILTERING

In the previous section signal models determined by the morphologically filtered spectra of the FFT based DCTC/DCS static feature vectors were evaluated. Results of those experiments showed that morphological filtering improved recognition performance by broadening the spectral peaks and smoothing of the noisy low level valley regions. For the multi-condition training data the performance of the new recognizers achieved a reduction in word error

rate of 14.72% over the base line MFCC features computed by the standard WI007 MFCC front-end, and a reduction in word error rate of 36% over the FFT based DCTC/DCS features presented in Section 5.3.2. Thus, experiments presented in Section 6.4 demonstrated the performance improving capability of the morphologically filtered DCTC/DCS features, which produce more a noise robust signal representation.

In order to further test the hypothesis stated in section 5.1, that recognizer performance would improve if the spectral peaks were not only preserved but also emphasized or enhanced, experiments with morphological filtering of the spectra of the LP based DCTC/DCS features were performed. In each of the following experiments 39 LP based DCTC/DCS features were computed from the morphologically-filtered spectra of the 13 component static feature vectors determined by LP-DCTC analysis. All other processing parameters, including the HMM recognizer, were identical to parameters of the evaluations presented in previous sections. Sections 6.5.1 and 6.5.2 present the evaluations discussed above. In both cases results are compared to the performance of the morphologically filtered FFT based DCTC/DCS features obtained by experiments presented in section 6.4.

6.5.1 MORPHOLOGICALLY FILTERED LP-DCTC/DCS SPECTRUM: MULTI-CONDITION TRAINING

The experiments presented in this section were designed to evaluate recognizer performance determined by the signal model computed from morphologically filtered spectra of the LP based DCTC/DCS static feature

vectors. Performance is compared to those of the recognizers determined by the 39 component feature vectors computed directly from FFT spectra, as reported in section 6.4. Additionally, the experiments presented in this section were performed with fixed block length. The morphological filter length was varied from 0 to 400 Hz, and the order of the linear prediction analysis was varied over a range of 0 to 100. Recall that an LP order of zero results in the FFT based DCTC/DCS signal analysis. Experimental results are presented in Figure 26.

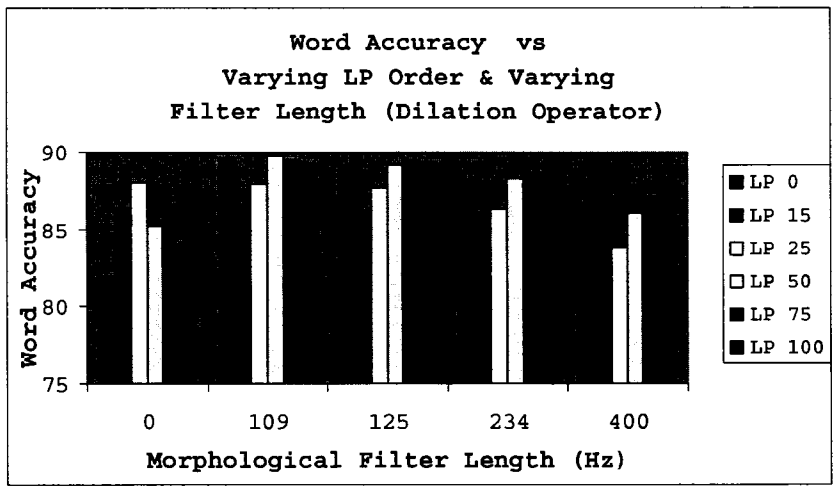

Figure 26 Performance of recognizers determined by varying the LP order and the morphological filter length (dilation operator), and evaluated with multi-condition training.

As indicated in Figure 26 recognizer performance for LP order 50 with filter lengths 109 and 125 Hz are quite close with a relative difference of only 0.06%. However, the best performance was achieved with morphological window length of 109 Hz resulting in a word error of 10.24%. Thus, the addition of peak sharpening by LP analysis resulted in a reduction in word error rate of 1.82%. Word accuracies for the recognizer determined by LP order 50, block length 11, and filter length 109 Hz are presented in Table 17.

The best performance achieved by recognizers determined by the same LP order but with morphological filter window width of 234 Hz achieved 11.77% WER, and the recognizer determined by a morphological window length of 400 Hz achieved 13.95% WER, which supports the theory stated earlier regarding extremely long filter length.

Table 17 Word Accuracy obtained with (dilation) filter length 109 Hz and block length 11. Multi-Condition Training.

SNR	**Word Accuracy (%)**			
	Test Set A	Test Set B	Test Set C	Average
Average	90.94	88.78	89.27	89.76

The improved performance was a result of morphological filtering with the addition of the 50^{th} order LP analysis, which produced sharper spectral peaks. These parameters indicate that morphologically filtered peaks produced by the LP based DCTC/DCS analyses are more pronounced than those produced by the FFT based DCTC/DCS signal model. Hence, for multi-condition training the combination of the high order LP analysis with morphological filtering resulted in a slightly more robust signal model, which improved recognizer performance.

6.5.2 MORPHOLOGICALLY FILTERED LP-DCTC/DCS SPECTRUM: CLEAN-CONDITION TRAINING

Section 6.4.2 presented the evaluation of recognizers determined from FFT based DCTC/DCS signal models, which were trained with the Aurora 2.0 clean-condition training data. The best performance was achieved with the dilation operator using a filter length of 109 Hz. This section presents the same

evaluation but the signal model was determined from morphologically filtered spectra of the LP based DCTC static coefficients.

Evaluations presented in this section were designed to determine the LP order and morphological filter window which would determine the best recognizer performance when training with clean-condition data. Therefore, preliminary experiments were performed to determine the best block length (BL=13) for clean-condition training with the LP based DCTC/DCS signal model. Once the block length was determined the evaluation was performed with the same experimental setup used for the experiments presented in Section 6.4.2. Results are presented in Figure 27.

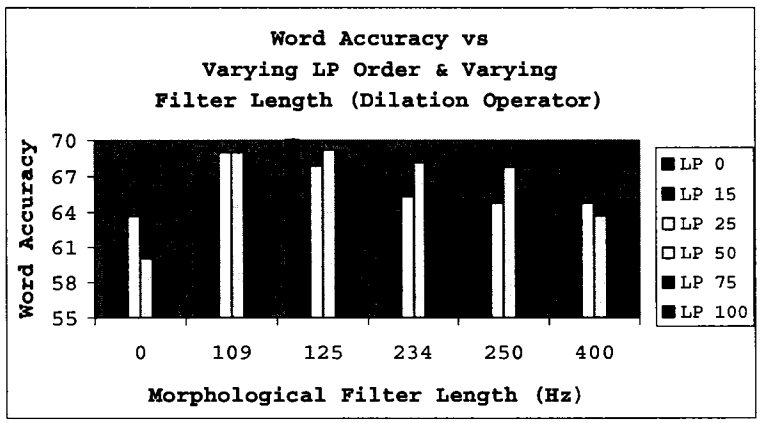

Figure 27 Performance of recognizers determined by varying LP order and morphological filter length (dilation operator), and evaluated with clean-condition training data.

In contrast to the evaluations with multi-condition training data the addition of LP analysis did not result in improved recognizer performance. Thus, the recognizer determined with LP order 0 and morphological window length 125 Hz still produced the best performance.

6.6 MORPHOLOGICAL FILTER TYPES

Thus far the objective of the experiments presented in this dissertation has been to improve Automatic Speech Recognition in noise. Pilot experiments determined that the dilation and close-open operators would be most likely to determine the most robust signal models. The close-open operator performs marginally better than the dilation operator but is three times more computationally demanding. Therefore, the dilation operator of morphological filtering was the only filter used in the experiments presented to this point. However, for completeness the other morphological filters were each tested with the LP based DCTC/DCS signal representation. The results of those evaluations are presented in this section.

EROSION OPERATION

The essential characteristic of the erosion operation is to reduce signal peaks, as was seen in Figure 21. Therefore, it was expected that the erosion operation would result in degraded recognizer performance. Figure 28 shows performance of recognizers determined from morphologically filtered (erosion) spectra of the LP based DCTC static components. As the LP order was increased the recognizer performance was further degraded. This behavior can be attributed to the fact that subsequent morphological filtering reduced sharp peaks produced by the LP analyses. This reduction of signal peaks results in the loss of important speech information and naturally produces a poor signal representation.

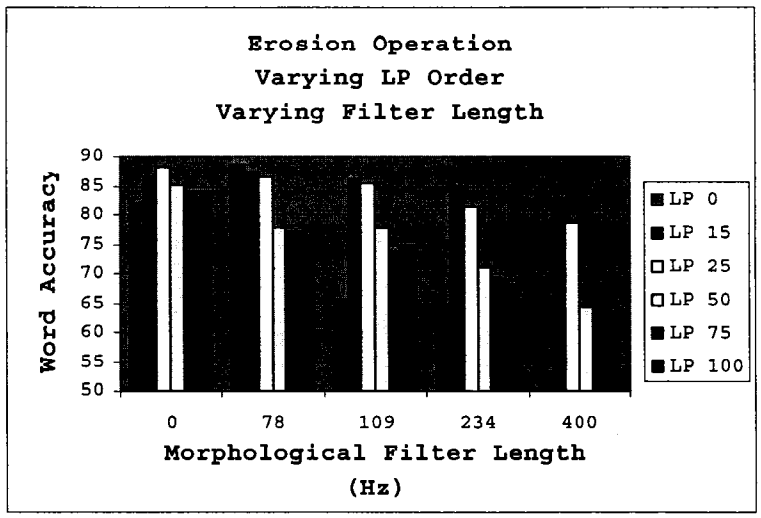

Figure 28 Erosion operation, varying LP order and varying filter length. Evaluation was performed with multi-condition training.

OPEN OPERATION

Figure 29 shows performance of recognizers determined from morphologically filtered (open) spectra of the LP based DCTC static components.

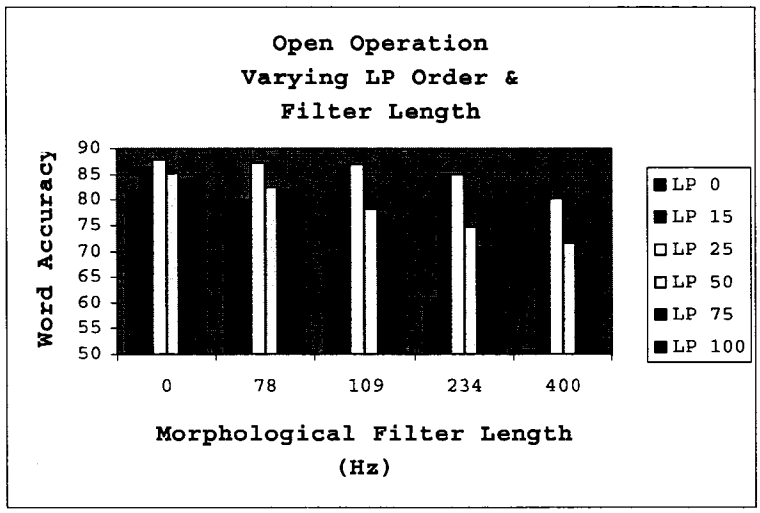

Figure 29 Open operation, varying LP order and varying filter length. Evaluation performed with multi-condition training.

As with the erosion operator, recognizer performance degraded as the LP order was increased

CLOSE OPERATION

As can be seen in Figure 30 the recognizer determined by the close operation performed slightly better than the one determined with the dilation operation. However, the close operation requires more than twice the computational demand for only a slight improvement in recognizer performance. Thus, the decision was made to use the dilation operation for most of the evaluations presented in this dissertation.

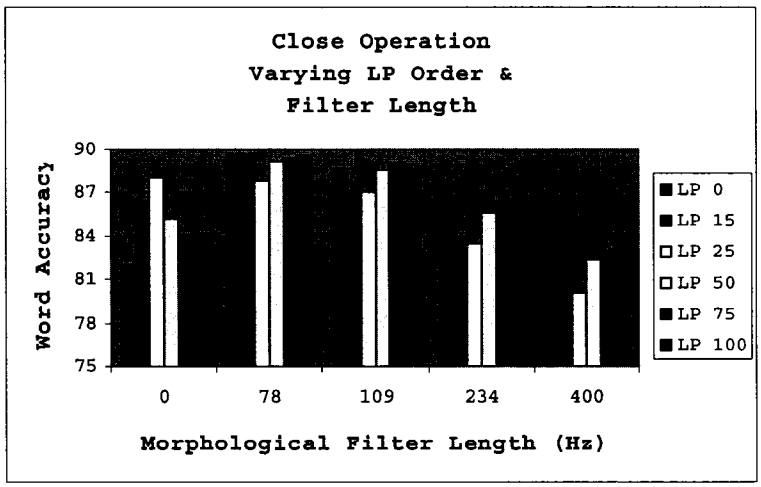

Figure 30 Close operation, varying LP order and varying filter length. Evaluation performed with multi-condition training.

OPEN-CLOSE OPERATION

The open-close operator performs quite well without LP analysis (LP order=0). Recall that the open-close operator begins with the open operation which essentially clips noisy impulse peaks, where the close stage of the operator emphasizes remaining peaks. Evaluations of the recognizers determined from the open-close operation are presented in Figure 31. The figure shows performance degrading as the morphological filter length and the LP order

are increased. However, even for morphological filter length of 400 Hz the performance of the recognizers with low LP order are a minimum of 10% higher than those with LP order 50 and higher.

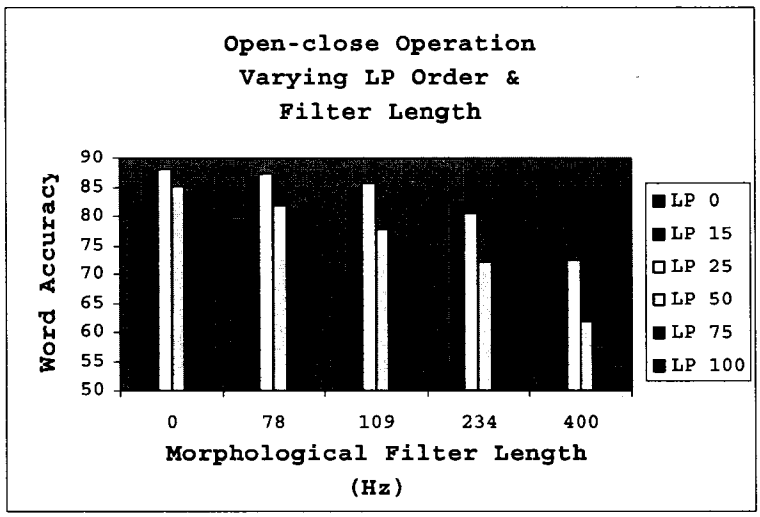

Figure 31 Open-close operation, varying LP order and varying filter length. Evaluation performed with multi-condition training.

CLOSE-OPEN OPERATION

As can be seen in Figure 32 the close-open operation performs very much like the dilation operator. However, filter lengths for the dilation operator which produces good signal models are not necessarily good choices for the close-open operator due to the cascading the dilation and erosion operators.

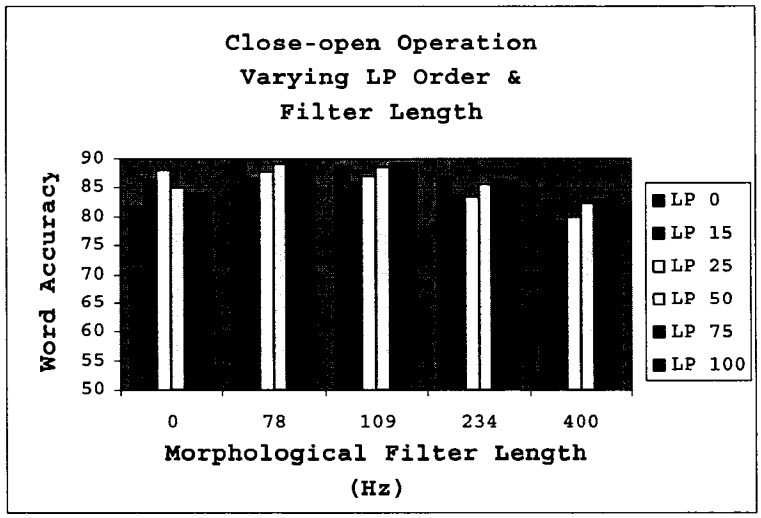

Figure 32 Close-open operation, varying LP order and varying filter length. Evaluation performed with multi-condition training.

To this point each of the experiments presented have been performed using the standard Hidden Markov Model recognizer described in Section 3.4. This recognizer configuration was determined by the ETSI Aurora Working Group so that front-end analysis algorithms could be compared as directly as possible. Thus, the use of the standard HMM recognizer configuration allows researchers to compare speech recognition results without the added complexity of determining whether changes in performance were due to improved speech signal representation or from the HMM configuration. However, the ETSI working group later determined that the original standard HMM configuration was not optimal, and could be improved.

In an attempt to provide a better baseline a second, more complex HMM recognizer configuration was later designed [59]. This new configuration is discussed in Section 6.7. The new HMM configuration was tested using the best parameters determined from experiments presented in Section 6.5. These

parameters were used as a starting point for determining the speech signal representation for the new HMM configuration. Results from these experiments are also presented in section 6.7.

6.7 RECOGNIZER ARCHITECTURE: INCREASED COMPLEXITY

Recall from section 3.5 that the original HMM architecture used one 18 state word model for each digit, which were modeled by one continuous density HMM with Gaussian probability density functions (referred to as mixtures) and a diagonal covariance matrix. After 16 iterations of training the final word models contained 6 Gaussian mixtures and the silence models contained 10 Gaussian mixture components. The configuration of the new HMM recognizer is described in the following section.

6.7.1 NEW HMM RECOGNIZER CONFIGURATION

The new HMM configuration uses the same basic initialization scheme as the previous HMM configuration. Thus the word models are initialized to 18 states with 1 Gaussian mixture, and initial parameters are determined by equally segmenting each utterance in the database, and computing the global mean and variance. However, an additional iteration of parameter estimation was added before the silence and short-pause model initialization. After the initialization of the silence models there is again an additional iteration of re-estimation of the model parameters. Also, the Viterbi algorithm is used to align the training data before further iterations of parameter estimation. As with the previous HMM configuration, additional Gaussian mixture components are added in between

parameter estimations. However, in this case the final word models contain 20 Gaussian mixtures and the final silence models contain 10 Gaussian mixtures. Furthermore there are a total of 103 training iterations where the previous HMM recognizer was trained with only 16 iterations.

6.7.2 EXPERIMENTS: THE NEW HMM CONFIGURATION

The close-open operation was expected to provide increased noise suppression over the dilation operation, which would result in a more noise robust signal representation. However, pilot experiments determined that the performance improvements achieved by filtering with the close-open operation were not enough to justify its use due to the significant increase in computational demand. Therefore, the dilation operator was used in each of the experiments presented in Chapter 5, Section 6.4, and Section 6.5. Experiments using the new HMM configuration require approximately 24 hours of computation time as compared to approximately 7 hours for the old HMM recognizer configuration. Whereas the computational demand resulting from filtering with the close-open operation and training the new HMM recognizer requires approximately 26 hours. This is only an 8.33% increase in performance from over 24 hours. Therefore the new HMM configuration was tested with the close-open operation.

The evaluation was performed with the Aurora 2.0 multi-condition training data. Block length was varied from 10 to 17, and LP order was varied from 0 to 100. Also, preliminary experiments showed the close-open filter length of 79 Hz resulted in more robust signal representation. Therefore, the evaluations were

performed with this parameter value. Recognizer performances are presented in Figure 33.

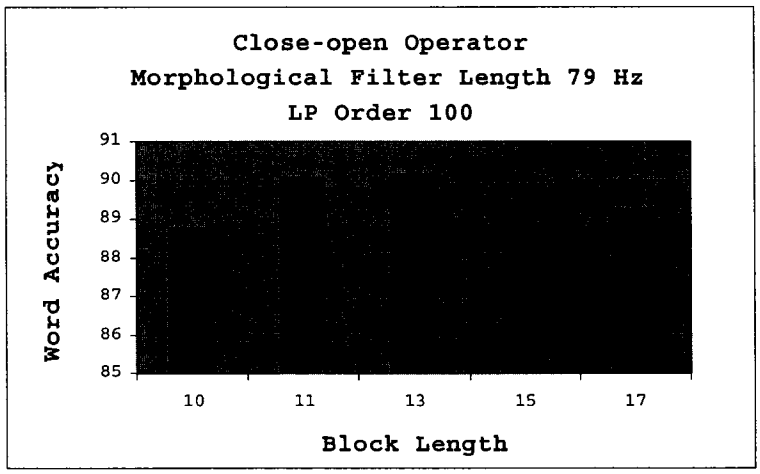

Figure 33 Recognizer performances for the close-open operation with window length 79 Hz, varying block length. The evaluation was performed with multi-condition training.

The recognizer determined by LP order 100 and block length 13 achieved 9.87% WER. Thus, the new recognizer achieved relative improvement of 4.17% in word error rate over the best recognizer performance presented in Section 6.5, and a 19.30% reduction in WER over the baseline recognizer evaluated in the control experiment in Section 5.3, which achieved 12.23% WER.

6.8 MORPHOLOGICALLY FILTERED SPECTRA: THE AURORA 3.0 DATABASE

Each evaluation already presented has been performed using the Aurora 2.0 database. The Aurora 3.0 database was used in the evaluations of the enhanced signal representation presented in this section. One evaluation was performed using each language of the Aurora 3.0 database. Experimental

parameters were chosen based on the values which were used to determine the best performance in experiments presented in experiments presented in previous chapters. In each case the LP order was varied over the range 0 to 75, block lengths 11 and 13 were tested, and the morphological filter length was varied over the lengths 78 Hz, 109 Hz, and 125 Hz. The results of the recognizers with the best performances are presented in Tables 18 through 21.

Table 18 Evaluation of the recognizer determined with the dilator operator with morphological filter length 78 Hz, LP order 25, block length 13, and the Spanish language database.

SNR Level	**Word Accuracy (%)**			
	Well-Matched	Medium-Mismatch	High-Mismatch	Average
Average	87.96	77.90	54.98	76.19

Table 19 Evaluation of the recognizer determined with the dilator operator with morphological filter length 109 Hz, LP order 25, block length 13, and the Finnish language database.

SNR Level	**Word Accuracy (%)**			
	Well-Matched	Medium-Mismatch	High-Mismatch	Average
Average	91.76	67.99	49.26	72.82

Table 20 Evaluation of the recognizer determined with the dilator operator with morphological filter length 109 Hz, LP order 0, block length 13, and the Danish language database.

SNR Level	**Word Accuracy (%)**			
	Well-Matched	Medium-Mismatch	High-Mismatch	Average
Average	79.03	53.95	38.17	60.04

Table 21 Evaluation of the recognizer determined with the dilator operator with morphological filter length 125 Hz, LP order 75, block length 11, and the German language database.

SNR Level	Word Accuracy (%)			
	Well-Matched	Medium-Mismatch	High-Mismatch	Average
Average	90.48	80.31	67.21	81.10

The ETSI published results and results from the literature were presented in Chapter 3. For convenience the ETSI published results and the best results from the literature are given again in Table 22. From Tables 18 through 21 it can seen that the evaluation of the recognizers determined with morphological filtering (dilator operator) result in performance better than the ETSI published results, except for the German language database for which performance is essentially equivalent. While the performances reported in Tables 18 through 21 do not exceed the best reported in literature they are important in that they are accomplished with only the addition of morphological smoothing. Each of the results reported in Table 22 (except for the ETSI Published) were achieved with several additional stages of signal processing. Possible future work could be to test morphological filtering with additional speech signal processing stages comparable to those used to obtain the best results reported in Table 22.

Table 22 Key Study Results for Aurora 3.0 Multi-Condition Training.

Author	Finnish	Spanish	German	Danish
ETSI Published	68.76	67.61	81.31	52.3
Bauerecker, H.	84.78	86.78	NA	74.44
Chen, C.	96.36	91.41	86.86	80.25
Droppo, J.	91.22	92.64	90.03	86.00
Andre, A.	91.42	94.48	92.88	94.57

6.9 CHAPTER CONCLUSIONS

The best recognizer performance was achieved by training with the new HMM recognizer with the signal model determined from morphologically filtering the spectra of the LP-DCTC derived static feature components. The improved recognizer achieved a reduction in word error rate of 19.30% over the performance of the recognizer determined from the MFCC based representation evaluated in the first control experiment presented in Section 5.3.1, and a reduction in word error rate of 39.80% over the FFT based DCTC/DCS speech signal representation with no morphological filtering, also presented in Section 5.3.1.

The best performance achieved with the original HMM recognizer configuration was 89.2%, and was obtained by the recognizer determined from the 39 component feature vectors, which were computed from the morphologically filtered spectra of the LP-DCTC static coefficients. .The recognizer performance represents a reduction in word error rate of 34.11% over baseline speech signal representation presented in Section 5.3.1, and a

reduction in word error rate of 11.69% over the standard MFCC signal model determined by the WI007 front-end. Thus, the results of the experiments presented in this chapter support the hypothesis that machine recognition would improve with morphological enhancement of the LP based DCTC/DCS speech signal representation.

CHAPTER VII
CONCLUSIONS AND FUTURE WORK

Many aspects of the use of morphological filtering with LP based DCTC/DCS analysis have been investigated. In the case of the Aurora 2.0 multi-condition training data the proposed morphological filtering methods have proven to be beneficial over the typical MFCC method and FFT based DCTC/DCS methods. Furthermore, there is much room for improvement for the dilator, close, and close-open filters which may lead to significant performance improvements. The use of variable morphological filter window lengths might result in better performance. Since the objective of this work was concerned with establishing the basic noise suppression capabilities of morphological filtering the more advanced benefits of this method have not been addressed.

7.1 CONTRIBUTIONS

A straight forward technique for spectral smoothing via morphological filtering with LP-DCTC/DCS analysis has been proposed for use with continuous digit string recognition.

For convenience, the key study results listed in Chapter 3 are repeated in Tables 23 and 24. Our best results including LP analysis are given in Tables 25 and 26, and best results without LP analysis are given in Tables 27 and 28. As can be seen in Table 23 the best result reported in literature for the Aurora 2.0 multi-condition training data was a 6.48% WER where our best achieved was a 9.87% WER. The WER achieved by C. Chen, et al was a relative reduction in

WER of 0.52% over our best. However, the performance achieved by Chen was achieved by mean subtraction followed by variance normalization of the static coefficients, and using a mixed auto-regression moving average filter after mean subtraction and variance normalization. Thus, there are three additional stages of signal processing as compared to our front-end analysis which adds only the morphological filtering. Additionally, D. Macho achieved a WER of 9.8% (as compared to our 9.87% WER) by two stages of filtering, one in the time domain and one in the frequency domain. The relative reduction of 0.7% WER was clearly achieved with an increase in computational demand.

Table 23 Key Study Results for Aurora 2.0 using Multi-Condition Training.

Author	**Aurora WI007**	**Cui, X.**	**Droppo, J.**	**Macho, D.**	**Chen, C.**
Test Set					
TSA	87.91	90.22	90.83	91.37	93.76
TSB	87.69	88.84	89.37	89.72	93.27
TSC	85.70	89.08	89.24	89.51	93.51
AVG	87.52	89.38	89.81	90.20	93.52

Table 24 Key Study Results for Aurora 2.0 using Clean Training.

Author	**Aurora WI007**	**Evans, W.D.**	**Kim, H.K.**	**Cui, X.**	**Chen, C.**
Test Set					
TSA	67.62	76.01	81.26	85.48	87.58
TSB	62.96	72.60	82.6	85.77	88.41
TSC	71.62	79.16	83.07	84.10	87.05
AVG	66.99	75.61	82.18	85.27	87.74

For the Aurora 2.0 clean-condition training data the best performance of 12.26% WER were again achieved by C. Chen, et al. This is a relative reduction of 1.44% in WER. Although this is a significant improvement over our WER it is

important to note that our results were achieved with minimal additional signal processing. Additionally, the primary objective of the work presented in this dissertation was focused on the multi-condition training data. It is possible that with additional signal processing the morphological filtering presented in this work would also further improve performance with the clean-condition training data.

Table 25 Best performances achieved with when evaluating with multi-condition training data.

SNR Level	Word Accuracy			
	Test Set A	Test Set B	Test Set C	Average
BL = 11 LP Order 50 Morphological Filter Length 109 Hz	90.94	88.78	89.27	89.76
BL=13 LP Order = 100 Advanced HMM Morphological Filter Length= 79	92.71	87.19	90.10	90.13

Table 26 Best performances achieved with LP Order = 50 when evaluating with clean-condition training data.

SNR Level	Word Accuracy			
	Test Set A	Test Set B	Test Set C	Average
BL = 13 Morphological Filter Length 125	70.332	65.25	73.07	69.23

Table 27 Best performances achieved with LP Order = 0 when evaluating with multi-condition training data.

SNR Level	Word Accuracy			
	Test Set A	Test Set B	Test Set C	Average
BL=10	86.39	82.25	81.06	83.61
BL=11	89.97	86.34	87.28	88.03
BL=11 Morphological Filter Length 109 Hz	90.76	88.61	89.02	89.57

Table 28 Best performances achieved with LP Order = 0 when evaluating with clean-condition training data.

SNR Level	Word Accuracy (%)			
	Test Set A	Test Set B	Test Set C	Average
BL=13 Morphological Filter ength 125 Hz	71.35	66.39	73.14	70.06

The highest performance reported in the literature was accomplished with significant signal processing. Our recognition accuracy on test data was comparable to the second highest result reported on the Aurora 2.0 multi-condition data with straightforward signal shaping, which requires much less signal processing than that used for the best reported results obtained with Aurora 2.0.

The original hypothesis was that morphological filtering would improve upon the results achieved by peak sharpening via LP-DCTC/DCS analysis demonstrated by experiments presented in Chapter 5. Experiments presented in Chapter 6 showed that morphological filtering did improve results achieved with LP-DCTC/DCS peak sharpening. By comparing results reported in Tables 25 through 26 it can be seen that the same experiments also showed that

morphological filtering without the benefit of LP-DCTC/DCS analysis achieved nearly equivalent results. In some cases the results achieved with the combination of LP-DCTC/DCS analysis and morphological filtering were improved but the improvements were at a cost of significant computational demand due to the LP analysis. Thus, morphological filtering provided more benefit than was originally predicted since noise robustness can be increased without additional significant computation demand.

The morphological filtering method was illustrated with a collection of experiments that were conducted using features computed using various morphological operators for spectral smoothing. Performance for the dilator operator was comparable to some of the best results in the literature while computational complexity was simpler and time requirements were lower.

7.2 FUTURE WORK

There are several suggestions for further research as follows:

1. Voice activity detection should be implemented with frame dropping dependent on the voice activity detection results.
2. Feature normalization, such as mean and variance normalization, can be applied to the final features after the morphological filtering.
3. Variable block spacing based on changes in spectral change and variable block size should be investigated.
4. Instead of a conventional HMM, a combination Neural Network/HMM could be employed.

REFERENCES

[1] Hansen, J.L., Clements, M.A., "Constrained iterative speech enhancement with application to speech enhancement", IEEE Trans. Signal Proc., Vol. 39(4), pp. 795-805, 1991.

[2] Pellom, B.L., Hansen, J.H.L., "An improved (auto:I, lsp:t) constrained iterative speech enhancement for colored noise environments", IEEE Trans, ASSP, Vol. 6, pp. 573-579, 1998.

[3] Sarikaya, R., "Robust and Efficient Techniques for Speech Recognition", PhD Theses, Dept. ECE Duke Univ. 2001.

[4] Yapanel, U., Hansen, J.L., Sarikaya, R., and Pellom, B., "Robust Digit Recognition in Noise: An Evaluation Using the Aurora Corpus," Proc. Of Eurospeech, Aalbor, Denmark, 2001.

[5] Furui, S., "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum," IEEE Trans. ASSP, Vol. 34, pp. 52-59, 1986.

[6] Milner, B., "Inclusion of Temporal Information into Features for Speech Recognition," Proc. ICSLP, pp. 256-259, 1996.

[7] Kuwabara, H., "Temporal Effect on the Perception of Continuous Speech and a Possible Mechanism in the Human Auditory System," Proc. Eurospeech, pp. 713-717, Berlin, Germany, 1993.

[8] Hanson, B. and Applebaum, T., "Robust Speaker-Independent Word Recognition Using Static, Dynamic and Acceleration Features: Experiments with Lombard and Noisy Speech," ICASSP, Vol. 2, pp 857-860, 1990.

[9] R.P. Lippmann, Carlson, B.A., "Speech Recognition by Humans and Machines Under Noisy Conditions with Severe Channel Variability and Noise," MIT Lincoln Laboratory, SPIE Vol. 3077 pp 46-56, 1997.

[10] Lippmann, R.P., and Martin, E.A., "Multi-Style Training for Robust Isolated-Word Speech Recognition," in Proceedings IEEE International Conference on Acoustics Speech and Signal Processing, pp 705-708, 1987.

[11] Rose, R., "Environmental Robustness in Automatic Speech Recognition," Proc. ICSLP, pp 2321-2324, Jeju Island, Korea, 2004.

[12] Cooke, M., Green, P., Josifivski, L., and Vizinho, A., "Robust Automatic Speech Recognition with Missing and Unreliable Acoustic Data," Speech Communication, Vol. 34(3), pp 267-285, 2001.

[13] Markel, J. and Gray, A., "Linear Prediction on Speech," Springer-Verlag, New York, NY, 1980.

[14] Rabiner, L. and Juang B., "Fundamentals of Speech Recognition," Prentice Hall, New Jersey, 1993.

[15] Picone J., "Signal Modeling Techniques in Speech Recognition," IEEE Proceedings, Vol. 81, pp. 1215-1247, 1993.

[16] Stevens S., "On the Psychophysical Law," *Psychol Rev* Vol. 64(3), pp 153-181 (1957). PMID 13441853.

[17] Nossair, Z. and Zahorian, S., "Dynamic Spectral Shape Features as Acoustic Correlates for Initial Stop Consonants," J. Acoust. Soc. Amer., Vol. 89, pp. 2978-2991, 1991.

[18] Baum, L., "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of a Markov process," Inequalities, Vol.3, pp. 1-8, 1972.

[19] Zahorian, S. and Jagharghi, A. "Spectral-shape Features versus Formants as Acoustic Correlates for Vowels," J. Acoust. Soc. Amer., Vol. 94, pp 1966-1982, 1992.

[20] Zahorian, S., Qian, D. and Jagharghi, A., "Acoustic-phonetic Transformations for Improved Speaker-independent Isolated Word

Recognition," Proc. ICASSP 91, pp. 561-564, Toronto, Canada, 1991.

[21] Bourlard, H. and Dupont, S., "Subband-Based Speech Recognition," Proc. ICASSP 97, Munich, Germany, 1997.

[22] Brown, P., "The Acoustic-modeling Problem in Automatic Speech Recognition," Ph.D. thesis, Computer Science Department, Carnegie Mellon University, 1987.

[23] Hansen, J.L., "Analysis and Compensation of Speech under Stress and Noise for Environmental Robustness in Speech Recognition," Speech Communication Vol. 20, pp 151-173, Elsevier.

[24] Rabiner, L.R., Cheng, M.J., Rosenberg, A.E., and McGonegal, C.A., "A Comparative Performance Study of Several Pitch Detection Algorithms," IEEE Trans Audio Electroacoust., Vol. ASSP-24, pp.399-417, 1976.

[25] De Cheveigne, A., and Kawahara, H., "YIN, A Fundamental Frequency Estimator for Speech and Music," J. Acoust. Soc. Am, Vol. 111, pp. 1917-1930, 2002.

[26] Shimamura, T., and Kobayashi, H., "Weighted Autocorrelation for Pitch Extraction of Noisy Speech," IEEE Trans. Speech Audio Processing, Vol. 9, No. 7, 2001.

[27] Charpentier, F., J., "Pitch Detection Using the Short-term phase spectrum," Proc. ICASSP, Tokyo, 1986.

[28] Flanagan, J.L, and Golden, R.M., "Phase Vocoder," The Bell System Technical J. Vol. 45f, pp. 1493-1509, 1966.

[29] Deng, L. and Sameti, H., "Transitional Speech Units and their Representation by Regress Markov States: Application to Speech Recognition," IEEE Trans. SAP, Vol. 4, pp. 301-306, 1996.

[30] Talkin, D., "A Robust Algorithm for Pitch Tracking," Speech Coding and Synthesis, pp. 495-518. Elsevier Science, Amsterdam, 1995.

[31] Atake, Y., Irino, T., Kawahara, H., Lu, J., Nakamura, S., and Shikano, K.,

"Robust Fundamental Frequency Estimation Using Instantaneous Frequencies of Harmonic Components," Proc ICSLP, Vol. II, pp. 907-910, 2000.

[32] Bergman, A.S., "Auditory Scene Analysis – The Perceptual Organization of Sound," MIT Press, Cambridge, 1990.

[33] Hess, W., "Pitch Determination of Speech Signals," Springer-Verlag, Berlin, ITU-T (1994), "ITU-T Recommendations, pp.11, 1994.

[34] Kawahara, H., Masuda-Katsuse, I., and de Cheveigne, A., "Restructuring Speech Representations Using a Pitch-adaptive Time-frequency Smoothing and an Instantaneous-frequency based F0 Extraction: Possible Role of a Repetitive Structure in Sounds," Speech Communications, Vol. 27, Nos. 3-4 pp. 187-207, Elsevier, Amsterdam, 1990.

[35] Liu, D., and Lin, C., "Fundamental Frequency Estimation based on the Joint Time-frequency analysis of Harmonic Spectral Structure," IEEE Trans. Speech Audio Processing, Vol. 9, No. 6, 2001.

[36] Nakatani, T., and Okuno H., G., "Harmonic Sound Stream Segregation Using Localization and its Application to Speech Stream Segregation," Speech Communications, Vol. 27, No. 3, pp. 209-222(14), Elsevier, Amsterdam, 1999.

[37] Hess, W., "Pitch and Voice Determination," in Advances in Speech Signal Processing, New York: Marcel Dekker, 1991.

[38] Secrest, B.G., and Doddington, G.R., "An Integrated Pitch Tracking Algorithm for Speech Systems," in Proc. Int. Conf. Acoust. Speech Sign. Process. (Boston), pp. 1352-1355, IEEE, New York, NY, 1983.

[39] Kawahara, H., Masuda-Katsuse, I., and de Cheveigne, A., "Restructuring Speech Representations Using a Pitch Adaptive Time-Frequency Smoothing and an Instantaneous-Frequency Based F0 Extraction: Possible Role of a Repetitive Structure in Sounds," Speech

Communications, 1998.

[40] Oppenheim, A., Schafer, R., Buck, R., "Discrete-Time Signal Processing," Prentice Hall Signal Processing Series, Upper Saddle River, New Jersey, 1998.

[41] Amari, S., "A Theory of Adaptive Pattern Classifiers," IEEE Trans. On Elec. Com., Vol. EC16, pp. 279-307, 1967.

[42] Parker, D., "Invention Report S81-64", File 1, Office of Technology Licensing, Stanford University, 1982.

[43] Parker, D., "Learning Logic," Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985.

[44] Rosenblatt, F., "Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms," Spartan Books, Washington, 1962.

[45] Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "Learning Internal Representations by Error Propagation," in Parallel Distributed Processing, Vol. 1, pp. 318-362. MIT Press, Cambridge MA. 1986.

[46] Werbos, P., "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," PD Thesis, Harvard University, Cambridge, MA. 1974.

[47] Widrow, B., and Hoff, M., "Adaptive Switching Circuits," Technical Reports 1553-1, Stanford University, Electron. Labs., Stanford, Ca, 1960.

[48] Baker, J.K., "The Dragon System: An Overview," IEEE Trans. Acoustics, Speech Signal Proc., ASSP-23 (1), pp. 24-29, 1975.

[49] Jelinek, F., "A Fast Sequential Decoding Algorithm Using a Stack," IBM J. Res. Develop., Vol. 13; pp. 675-685, 1969.

[50] Bahl, L.R., Jelinek, F., "Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition," IEEE Trans.

Information Theory," IT-21: 404-411, 1975.

[51] Jelinek, F., Bahl, L.R., and Mercer, R.L., "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," IEEE Tr4ans. Information Theory, IT-21: 250-256, 1975.

[52] Jelinek, F., "Continuous Speech Recognition by Statistical Methods," in Proc. ASA Meeting, Washington, D.C., 1976.

[53] Bakis, R., "Continuous Speech Word Recognition via Centisecond Acoustic States," in Proc ASA meeting, Washington, D.C., 1976.

[54] Jelinek, F., Bahl, R.L., and Mercer, R.L., "Continuous Speech Recognition: Statistical Methods," in Handbook of Statistics, II, P.R. Krishnaiad, Ed. Amsterdam, The Netherlands: North-Holland, 1982.

[55] Bahl, L.R., Jelinek, F., and Mercer, R.L., "A Maximum Likelihood Approach to Continuous Speech Recognition," IEEE Trans. Pattern Anal. Machine Intell., PAMI-5: 179-190, 1983.

[56] Morgan, N., Bourlard, H., "An Introduction to Hybrid HMM/Connectionist Continuous Speech Recognition," IEEE Signal Processing Magazine, pp. 25-42, 1995.

[57] Hirsch, H., Pearce, D., "The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions," ISCA Archive, ICSLP, Beijing, China, 2000.

[58] Jelinek, F., "Statistical Methods for Speech Recognition", the MIT Press, Cambridge, Massachusetts, 1999.

[59] Pearce, D., "Enabling New Speech Driven Services for Mobile Devices: An Overview of the ETSI Standards Activities for Distributed Speech Recognition Front-ends," Applied Voice Input/Output Society Conference (AVIOS2000), San Jose, Ca, 2000.

[60] ITU recommendation G.712, "Transmission Performance Characteristics of Pulse Code Modulation Channels," 1996.

[61] ETSI standard document, "Speech Processing, Transmission and Quality aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; compression algorithm," ETSI ES 201 108 v1.1.1 (200-02), 2000.

[62] Cui, X., Markus, I., Zhu, Q., Alwan, A., "Evaluation of Noise Robust Features on the Aurora Databases," ICSLP, pp. 481-484, Denver, Colorado, 2002.

[63] Macho, D., Mauuary, L., Noe, B., Cheng, Y., Ealey, D., Jouver, D., Kelleher, H., Pearce, D., Saadoun, F., "Evaluation of a Noise-Robust DSR Front-end on Aurora Databases," ICSLP, pp.17-20, Denver, Colorado, September 16-20, 2002.

[64] Hayes, M.H., "Statistical Digital Signal Processing and Modeling," John Wiley & Sons, Inc., Indianapolis, Indiana, 1996.

[65] Bauerecker, H., Climent, N., Padrell, J., "On the Advantage of Frequency-filtering Features for Speech Recognition with Variable Sampling Frequencies. Experiments with Speech-Dat Car Databases," Proc. EUROSPEECH, pp. 869-872, Geneva, 2003.

[66] Chen, C., Filali, K., Bilmes, J., "Front-end Post-processing and Back-end Model Enhancement on the Aurora 2.0/3.0 Databases," ICSLP, pp. 17-20, Denver, Colorado, 2002.

[67] Macho, D., Nadeu, C., Herando, J. Padrell., J., "Time and Frequency Filtering for Speech Recognition in Real Noise Conditions," Proc. Workshop on Robust Methods for Speech Recognition in Adverse Conditions, pp. 111-114, Tampere, Finland, 1999.

[68] Andre, A., Burget, L., Dupont, S., Garudadr, H., Grezl, F., Hermansky, H., Jain, P., Kajarekar, S., Morgan, N., Sivadas, S., "Qualcomm-ICSI-OGI Features for ASR," in Proc. Int. Conf. on Spoken Language Processing. Denver, Colorado, 2002.

[69] Kim, H., Rose, R., "Evaluation of Robust Speech Recognition Algorithms

for Distributed Speech Recognition in a Noisy Automobile Environment," ICSLP, pp. 233-237, Denver, Colorado, 2002.

[70] Evans, N., Mason, J., "Computationally Efficient Noise Compensation for Robust Automatic Speech Recognition Assessed Under the Aurora 2/3 Framework," ICSLP, pp. 485-488, Denver, Colorado, 2002.

[71] ETSI-SMG Technical Specification, "European Digital Cellular Telecommunication System (Phase I); Transmission Planning Aspects for the Speech Service in GSM PLMN System," GSM03.50, version3.4.0, July 1994.

[72] "ITU-T SG16 Multimedia Terminals, Systems, and Applications," ITU-T Software Tools Library.

[73] Droppo J., Deng, L., Acero, A., "Evaluation of SPLICE on the Aurora 2 and 3 Tasks," in Proc. Int. Conf. on Spoken Language Processing. Denver, Colorado, 2002.

[74] Maragos, P., Schafer, R., "Morphological Filters-Part I: Their Set Theoretic Analysis and Relations to Linear Shift Invariant Filters," Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 8, 1987.

[75] Serra, I., "Image Analysis and Mathematical Morphology," New York Academic, 1982.

[76] Gallagher, N.C., Wise, G.L., "A Theoretical Analysis of the Properties of the Median Filter," IEEE Trans. Acoust. Speech, Signal Proc., Vol. 29, No. 6, pp. 1136-1141, 1981.

[77] Justusson, B.I., "Median Filtering: Statistical Properties," Two Dimensional Digital Signal Processing II: Transforms and Median Filtering," T.S. Huang, Editor. Berlin: Springer-Verlag, pp. 161-196, 1981.

[78] Sternberg, S.R., "Biomedical Image Processing," IEEE Computer Magazine, pp. 22-34, 1983.

[79] Goetcheran, V., "From Binary to Graytone Image Processing Using Fuzzy Logic Concepts," Pattern Recognition, Vol. 12, pp. 7-15, 1980.

[80] Rodenacker, K., Gais, P., Jutting, U., Burger, G., "Mathematical Morphology in Grey Images," in Proc. European Signal Processing Conference, 1983.

[81] Marple, S. Jr., "Digital Spectral Analysis with Applications," Prentice-Hall, Englewood Cliffs, NJ, 1987.

[82] Atal, B.S., Hanar, L.S, "Speech Analysis and Synthesis of by Linear Prediction of the Speech Wave," J. Acoust. Soc. America, Vol. 50, No. 2, pp. 637-655, 1971.

[83] Young, S., Odell, J., Ollason, D., Valtchev, V., Woodland, P., "The HTK Book," Cambridge University, Cambridge, 1995.

[84] Heeman, P., Hosom, J., Yonghong, Y., "Automatic Speech Recognition at CSLU," http://cslu.cse.ogi.edu/asr.

[85] Kokkinos, I., Maragos, P. "Nonlinear Speech Analysis Using Models for Chaotic Systems," IEEE Transactions on speech and Audio Processing, Vol. 13, No.6, pp. 1098-1109, 2005.

[86] TI-46 Word Speech Database, SPEAKER-DEPENDENT ISOLATED WORD CORPUS (TI46 CDROM), NIST Speech Disc 7-1.1, 1991.

APPENDIX A
HTK COMMANDS

HCOPY: SIGNAL ANALYSIS

Each of the HTK tools can parameterize waveforms. However, it is more efficient to compute the signal representation, or parameterize all of the data at once. Most of the parameterization presented in this dissertation was performed using the ODU Speech Communications Laboratory front-end analysis software, TFRONTM. However the WI007 baseline experiment was performed using the HTK HCOPY tool to determine the speech signal representation.

The HCOPY tool converts ("copies") one or more source files to an output file, using the specified parameterization. The input speech data can be in any supported format (Examples: NIST, SPHERE, HTK) but the output is always in HTK format.

HTK File format

Number of Samples	Number of samples contained in the waveform.
Sample Period	Frame Rate.
Number of Features	Number of parameters computed for each wave form.
Bytes per Sample	Number of bytes per sample.
Sample Type:	USER, MFCC, LPCC.

Sample HTK Header File

Sample Bytes:	156	Sample Kind:	USER
Num Comps:	39	Sample Period:	10000.0 us
Num Samples:	164	File Format:	HTK

Note: All times must be given in 100ns units and as floating-point numbers.

Inputs:

- Configuration file contains basic configuration information
- Sentence list file contains the list of (wave form files) sentences to process

Outputs:

- HTK format feature files, one file for each waveform file

Typical usage: **HCopy** –A –D –T 1 –C config_file –S sentence_list

The options -A, -D, -T, -C, and -S, are standard options, which are common across all of the HTK tools.

–A Signals HTK to print out the command line arguments and is useful for debugging.

-D Signals HTK to print the version of HTK.

–T Specifies the trace level for debugging. There are four trace levels for the HCOPY tool, specified using octal base numbers, 1, 2, 4, and 10. The flag value 1 specifies basic progress reporting, and is the one used for experiments presented in this dissertation.

–C Specifies the configuration file, in this example config_file, which is a text file containing configuration parameters for the HCOPY tool.

–S Specifies the wave form files for computing parameterizations. In this example sentence_list is a text file containing the list of wave form files

After HCopy is executed, one parameter file should have been created for each of the wave files specified in the sentence list.

HCOMPV: HMM INITIALIZATION

HCompV calculates the global mean and covariance of a specified set of training data, and uses these to initialize all mixture components of all models.

Inputs:

- Parameter files, for example those created with HCOPY

Outputs:

- A file that contain initial HMM models with each mixture component set equal to the overall training data global mean and covariance.

Typical usage: **HCompV** –A –T 2 –D –C config_file –o hmmdef –f 0.01 –m –S train_list –M hmm_dir proto

The –A, -T, -D, and –C options are as described for HCopy.

-o hmmdef is used to specify that the initialized HMMs are to be stored in a text file called hmmdef.

–f 0.01 sets the minimum variance to 0.01.

–m causes the means to be updated

-S train_list instructs HCompV to use the parameterizations stored in files listed in the text file train_list.

-M hmm_dir proto is used to specify hmm_dir as the output directory path name, and proto is a text file that contains the hmm proto-type.

For a system completely determined by HTK the parameter files are created with HCOPY.

HEREST: ITERATIVE TRAINING

The Baum-Welch algorithm HERest performs a single re-estimation of the parameters of the HMMs. For each training utterance a set of accumulators are updated simultaneously.

HEREST operates in two stages.

1. For each training file the accumulators for state occupation, state transition, means and variances are updated.
2. The accumulators are used to calculate new estimates for the HMM parameters.

Inputs:

- Features files
- Initial HMM configurations
- Setup information

Outputs:

- Updated HMM models

Typical usage: HERest -A –D –T 4 –C config_fil –I labels –t 250.0 150.0 1000 –S
train_list –H macros –H models –M hmm_dir hmm_list.

-A, -D, -T, -C, in this work, the same as in previous commands.

–I instructs HERest to load the master label file labels. The master label files contain transcriptions for each of the sentence files. These transcriptions include the silence and short pause symbols.

–t sets the pruning threshold to 250 so that during backward probability computation any log β values more than 250 below the maximum value are ignored. A pruning error causes the threshold value to be increased by 150.0. This process continues until the limit of 1000 is reached.

–H instructs HERest to load the model and macro files, respectively. Macros and models are files that contain model parameters, and are generally quite large.

–M sets the output directory for the current HMM model and macro files. Hmm_dir is the output directory for the models and macros

Hmm_lst is a text file containing the list of HMM models.

HHED: EDITING HMM MODELS

HHed is used to manipulate the HMM definitions. In this work it was used to ledit the HMMs and output a transformed set of HMMS. HHed syntax consists of a comma separated list of item sets. For example the command AT 1 3 0.1 {*.transP} will add a transition from state 1 to state 3 with probability 0.1.

Inputs:

- Macros
- Models
- Hmm list

Outputs:

- Edited HMMs

Typical usage: **HHed** –T 2 –H macros –H models -M output edit_file hmm_list.

The edit_file is a text file containing edit commands, and hmm_list is the set of HMMs to be edited.

–H instructs HHed to load the macro and model files, respectively.

–M instructs HHed to store the output model file in the specified output file.

HVITE: VITERBI BASED RECOGNITION

HVITE matches a speech file against a network of HMMs and outputs a transcription. A label file and dictionary are read in and used to create a model

based network. The expansion is determined automatically from the dictionary and hmm list.

Inputs:

- Configuration file
- Macros
- Models
- Dictionary file
- Word list
- Grammar specified in a text file called net
- Sentences to be tested, listed in the text file test_list

Outputs:

- Hypothesis transcriptions contained in a text file. In this example the output text file is called output.

Typical usage: **HVite** –A –D –T 1 –C config_fil –H macros –H models –w net –l "*" –i output dict word_list –S test_list.

This command causes HVite to then load the network file (net) and match it against each of the test files specified in test_list. The word_list file contains a list of the models.

–l "*" allows the output path to be appended to each label stored in the master label file.

–i option causes the HVite output to be stored in the specified master label file.

-w specifies the network file

HRESULTS: PERFORMANCE EVALUATION

HRESULTS is the performance analysis tool. It reads in a set of label files output from HVite and compares them with the reference transcription file for the corresponding sentence file. Recognition statistics are output for each sentence.

Inputs:

- Labels for test sentences
- The HMM list specified in hmm_list

Outputs:

- A text file containing recognition statistics

Typical usage: **HResults** –D –A –e ??? sp –e ??? sil –p –I labels hmm_list recognition_file.

The file specified by recognition_file contains the output transcriptions from HVite. HResults is applied to each file in recognition_file. The hmm_list file contains the list of each of the HMM models, and noise_labels contains the list of label files for each noise type.

–e ??? causes HResults to ignore sp and sil wherever they occur in the transcription files.

–p causes HResults to output the confusion matrix.

–I instructs HResults to load the label files for the test files.