Environmental Robustness in Speech Recognition using Physiologically-Motivated Signal Processing

Yoshiaki Ohshima

Submitted to the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy at

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

December 1993

List of Figures

List of Tables

Abstract

Environmental robustness is one of the most important factors that determines the success of a speech recognition system. Several different types of approaches to environmental robustness have been developed in recent years, including cepstral normalization to compensate for acoustical differences between training and testing conditions of a recognition system, the use of arrays of microphones to separate speech from noise sources arriving from different directions in space, and the use of signal processing schemes that are based on knowledge of the human auditory periphery.

This thesis examines methods by which speech recognition systems can be made more environmentally robust by analyzing the performance of a representative model of the auditory periphery developed by Seneff. The major goals of the thesis are threefold. First, we document to a greater extent than had been done previously the extent to which the Seneff model reduces the degradation in speech recognition accuracy incurred when testing conditions include additive noise and/or distortion introduced by linear filtering that was not present when the system was trained. Second, we examine the extent to which individual components of the nonlinear neural transduction stage of the Seneff model contribute to recognition accuracy by evaluating recognition with individual components of the model removed from the processing. Third, we determine the extent to which the robustness provided by the Seneff model is complementary to and independent of the improvement in recognition accuracy already provided by existing successful acoustical pre-processing algorithms such as the CDCN algorithm.

The Seneff model provides two types of outputs which describe the putative mean rate of neural activity in a narrow range of frequency and temporal synchrony of this activity to the incoming speech sound. These outputs can be regarded as different estimates of the spectral energy in the incoming speech.

It is found that both the mean-rate and synchrony outputs of the Seneff model provide better recognition accuracy than what is obtained using conventional signal processing using cepstral coefficients derived from linear prediction, both when speech is subjected to artificially-added noise and when speech is modified by unknown linear filtering. Although there are 40 frequency-specific mean-rate and synchrony outputs in the original Seneff model, we found that no loss in recognition accuracy is incurred if classification decisions are made on the basis of 5 principal components of the mean-rate outputs or 10 principal components of the synchrony outputs.

The neural transduction (NT) stage of the Seneff model consists of a cascade of components that perform half-wave rectification, short-term adaptation, lowpass filtering, and automatic gain control (AGC). Of these components, short-term adaptation appears to be the most important. Nevertheless, no component could be removed from the model without sacrificing recognition accuracy.

We develop several ways of combining auditory processing with conventional cepstral processing using environmental normalization techniques such as CDCN. This is done either by normalizing the cepstra derived from the outputs of the auditory model or by normalizing the input speech waveform directly. We show that the recognition accuracy provided by physiologically-motivated signal processing can (in some circumstances) be further improved by combination with environmentally-normalized cepstral processing.

In order to combine auditory processing with environmentally-normalized cepstral processing, we develop methods to resynthesize speech both from its cepstral representation, and from the outputs of the auditory model. The use of speech resynthesized from cepstral coefficients provides a modest improvement in recognition accuracy. The use of speech resynthesized from the outputs of the Seneff model does not improve accuracy, but it is intelligible and may be useful for speech enhancement.

Acknowledgments

First, I would like to thank Professor Richard M. Stern, my advisor, for his invaluable guidance, support, and encouragement. He has been not only an academic hero to me, by whom I was motivated into this area of research, but also as a mentor with a great deal of enthusiasm and patience.-He has let me pursue ideas and possibilities, and still has never lost faith in me even at the most difficult times.

I would also like to thank the thesis committee, Professor Vijaya Kumar, Professor Rick Carley, and Dr. Wayne Ward, whose insightful comments and criticism have made this work turn into what it is now.

CMU Speech Group has been the best environment for research, and among all of its members, I am indebted to Bob Weide and Eric Thayer for their knowledge and expertise in SPHINX speech recognizer and the speech database. In my early days, researchers and senior colleagues such as Fil Alleva and Alejandro Acero kindly helped me to acquaint me with the frontier of speech technology. Fellow graduate students of Rich Stern's to date, as well as in the recent past, Tom Sullivan, Fu-Hua Liu, Sammy Tao, and Pedro Moreno, have always been excellent critics of my ideas, and I have valued my philosophical discussions with them. It has been quite a learning experience for me to have opportunities to work with such bright minds.

I would also like to thank Dr. Stephanie Seneff and Dr. Hynek Hermansky for allowing me to use their front-end programs and also for comments on preliminary results. Dr. Roy Patterson has been kind enough to help me understand his model and to give me advice at times. I am thankful to Charles Jankowski for exchanging ideas and discussions as well as for providing us with babble speech data. I would like to thank Mike Phillips for his help for the use of MFCC front-end.

I am grateful to the people at IBM Japan and IBM Tokyo Research Laboratory for their financial support and continuous encouragement by providing the scholarship program. In particular, I am indebted to Mr. Masaaki Okochi for believing in the virtue of the proposed work and for allowing me to be absent from my colleagues at work for such a long time.

Finally, I would like to thank my family and friends for their love, support, and so much more. My parents, Ryohei and Tomoko Ohshima, have been wonderful people that I have always aspired to be like. My sister, Junko, has often helped me a lot by being a thoughtful companion to her elder sibling. I am very much grateful to Tom Sullivan, John Hampshire, Prithvi Rao, and Alexei Sacks for being best friends of mine by understanding what I am and for having been around whenever I needed them. For the same reason, I am sincerely grateful to my dance partner and best friend, Cindy Wood. But for her wisdom, friendship, courage and sweet personality, I would have not been able to complete this work by getting over the toughest moments that occurred often during the last several months.

Chapter 1 Introduction

The development of real time speech recognition systems which are capable of handling large-vocabulary, speaker-independent, continuous speech has been one of the major technological challenges in speech research. CMU's SPHINX[1] speech recognition system is one of the best systems that has been developed to meet these goals. The development of successful high-end speech recognition systems has enabled us to anticipate the practical use of spoken language systems in the near future for a number of application areas. This thesis is concerned with methods by which speech recognition systems and spoken language systems can be made more environmentally robust.

It is very often the case that a speech recognition system shows an excellent recognition rate in a feasibility study conducted in the laboratory, but fails to maintain similar performance in real and uncontrolled environments such as offices using a desk-top microphone. The key element that makes the difference is the environment, or more specifically, the fact that the speech recognizer is not designed to deal with changes in environmental conditions. In the laboratory, a high-quality microphone is carefully placed near the speaker, the noise level is kept low, and casual access to the room is restricted during the session when the system is trained and tested. On the other hand, when the system is used in the field, a low cost microphone may be preferred, its placement with respect to the speakers may not be consistent, the amount of reflected echoes from the walls and the ceiling may no longer be negligible, air conditioning may turn on and off, unexpected telephone rings and typewriter noise may be heard, and people come in, chat, and leave at any moment. For the speech recognizer, all of these phenomena appear as degradations in the quality of input speech signal, for the system does not have the adaptive capability of distinguishing the speech from other acoustic phenomena and environmental perturbations. Lack of robustness with respect to environmental changes is a technical vulnerability to be overcome, as practical spoken language systems used in everyday life must be are capable of dealing with low signal-to-noise ratios, inconsistently and distantly placed microphones, the effects of reverberant room acoustics, and so on.

There are several different approaches that have applied to the problem of acoustical robustness for automatic speech recognition:

- The application of signal processing techniques using a model of a communication channel with additive noise and linear distortion
- Array processing techniques using more than one input transducer
- The use of physiologically-motivated models to mimic the noise robustness in the human auditory system

There has been extensive research using the first approach over the past several decades, in which researchers tried to “clean up” speech recorded in noisy environments, and to compensate for the effects of room acoustics for better intelligibility. Recently Acero[2] proposed a new method that jointly compensates for the effects of noise and linear filtering, and he demonstrated its usefulness by applying it to the SPHINX speech recognition system.

The second approach is based on signal processing techniques first developed in the area of beamforming for directional antennae. It is expected that traditional array processing techniques will provide some improvement in recognition accuracy in at least some environments, by enhancing the speech signal in the presence of interfering sources when the two have different directional or spatial characteristics. Further improvement may also be obtained by combining array processing with the first or the last approaches, which basically work with a single input channel.

The last approach is the one we would like to pursue in this thesis research. The motivation is the fact that the human auditory system adapts itself as the environment changes. Therefore, it is capable of dealing with various kinds of speech degradations gracefully, whereas ordinary speech recognition systems are not. Thus, we expect to be able to enhance environmental robustness by integrating useful features of auditory processing into the speech recognition system.

One might question the significance of the auditory periphery as the front end for the entire information processing of speech. For example, it may seem reasonable to believe that a considerable amount of high-level central processing is involved in the mechanics of understanding human speech, such as guessing phonetic cues buried in background noise. However, it should be noted that not only central processing but also peripheral processing plays an important role. Several attributes of physiological signal processing observed at the peripheral level seem to help improve robustness with respect to noise and other degradations, probably by providing a more useful representation of the input speech for later information processing at higher levels. The implication is that the output from the auditory periphery may preserve information that is relevant to the original message and yet less variable than the degraded input signal.

As we will discuss in the next chapter, auditory processing can be conceptually divided into three cascaded sections:

1. a bandpass-filter bank to model the frequency analysis of the cochlea,
2. nonlinear signal processing that simulates other peripheral functions, and
3. the construction of spectral images that provide alternative representation of the input signal to be used by higher centers.

Researchers have proposed models of the peripheral auditory system with different emphasis on various aspects of the physiology for more than a decade, and some have also conducted experiments in speech recognition which demonstrated the power of auditory modelling. Hunt[3] [4], for example, showed that a front end that incorporates an auditory model can work better than conventional front ends in additive Gaussian noise. Ghitza[5] also showed the superiority of a different physiologically-motivated model of auditory processing compared to a more conventional front end in digit-recognition tasks in noise. Meng[6] showed that the physiologically-motivated model of Seneff[7] achieved better performance than several types of conventional signal processing algorithms in the task of recognizing sixteen American vowels in both clean and noisy environments. It should be noted that these experiments only dealt with artificially-added white noise, and not the degradations to the signal experienced in realistic acoustical environments.

Despite these experimental results that have shown that signal processing algorithms based on auditory modelling can outperform conventional processing in various experimental conditions that simulate noisy environments, the use of physiologically-motivated peripheral signal processing is not yet a standard feature of speech recognition systems. Two possible reasons for this low popularity are:

1. Relatively little research has addressed the issue of exactly what enables successful auditory models to achieve their good performance
2. Auditory models are usually much more computationally demanding than conventional signal processing

In other words, while much attention has focussed on the development of auditory models that try to replicate various physiological phenomena, little is yet understood about what specific aspects of them are useful for the task of speech recognition. It is not our intention to study the extent to which auditory models can become an accurate replication of the human auditory system. Instead we are rather interested in their ability to produce features that we think are relevant to speech perception, and in studying them to look for better representations of speech which can be useful for speech recognition.

The goals of this project are to identify the most crucial signal processing components of the auditory modelling that has been used in the context of speech recognition, to evaluate the usefulness of such physiologically motivated front-end processing and the computational complexity for speech recognition, to compare the effectiveness of physiological models with that of conventional signal processing techniques, and to evaluate the extent to which the combination of auditory models and acoustical pre-processing can provide complementary benefits that produce greater improvements in recognition accuracy than what is possible by use of either technique alone.

So far we have reviewed the state of the art in speech recognition, and described robustness as an important issue. We have then described the general nature of the problems that we are trying to investigate, summarized a few of the approaches that have been used to deal with these problems, and defined our technical approach. From the next chapter on, the thesis describes the following topics:

Chapter 2 describes the SPHINX speech recognition system.

Chapter 3 addresses signal processing issues in the area of environmental robustness for speech recognition, and reviews acoustical preprocessing algorithms as one of the most effective solutions to the environmental robustness problems in the conventional signal processing approach.

Chapter 4 reviews previous auditory models as an alternative approach to the robustness issues. It identifies three major function blocks commonly found in the engineering models of auditory periphery, and analyzes each block's functionality in terms of robustness.

The following three chapters describe the experimental development of this research. Chapter 5 describes the baseline performance of physiologically-motivated signal processing in comparison with conventional signal processing in conjunction with acoustical preprocessing.

Chapter 6 describes the evaluation of individual nonlinear signal processing component of physiologically-motivated in term of its contribution to environmental robustness.

Chapter 7 describes various attempts to combine physiologically-motivated signal processing and acoustical preprocessing in the waveform and parameters domains. Waveform resynthesis/reconstruction techniques for the integration are explained.

Chapter 8 concludes the thesis providing the major contributions and a few suggestions for future work.

Chapter 2 The SPHINX Speech Recognition System

This chapter briefly describes CMU's SPHINX speech recognition system. A block diagram of the SPHINX front end is shown in Figure 2-1.

2.1. Front-End Signal Processing

The goal of front-end signal processing is to extract relevant feature parameters of speech, which are more suitable for the purpose of speech recognition than the input speech waveform itself in terms of information rate and reduction of redundancy. The SPHINX system[8] was originally developed using conventional signal processing techniques based on Linear Predictive Coding (LPC)[9]. A set of feature parameters called LPC cepstral coefficients (LPCC)[10] is computed based on an LPC analysis of a segments of speech. The stages of signal processing that produce the LPCC features in the conventional SPHINX LPCC front end are summarized as follows:

- Sampling at a frequency of 16 kHz
- Multiplication by a Hamming window of 20-msec duration, which is updated every 10 msec. Each 20-msec segment is referred to as a *frame* of speech.
- Preemphasis using a highpass filter (HPF) with transfer function $H(z) = 1 - 0.97z^{-1}$
- Extraction of 14 autocorrelation coefficients of the speech waveform in each frame
- Smoothing of autocorrelation coefficients by a lag window
- Computation of 14 LPC coefficients using Levinson-Durbin recursion
- Computation of 32 LPC cepstrum coefficients (LPCC) in linear frequency
- Production of 12 frequency-warped cepstral coefficients using a bilinear transform

As a result of front-end processing, the incoming speech waveform is transformed into a discrete-time sequence of LPCC parameters, each of which characterizes the corresponding short-term segment of speech called a frame. In frame-based analysis, it is assumed that parameters are statistically uncorrelated between frames. Frame-based parameters are sometimes called static features, and difference of parameters in time are also called dynamic features. In SPHINX, both cepstrum parameters and difference cepstra are used as features, as well as the energy and difference energy of each frame.

2.2. Vector Quantization

After LPCC parameters are computed, further reduction of data is achieved by using a technique called Vector Quantization (VQ). First, the mean locations of the feature space called centroids are computed by using an iterative clustering algorithm[11]. Feature parameters are represented by the identity of their closest centroid, *i.e.* the index number of the corresponding centroid. The set of centroids that discretize feature parameters into index symbols is specifically called a codebook, and the produced symbols are called codewords.

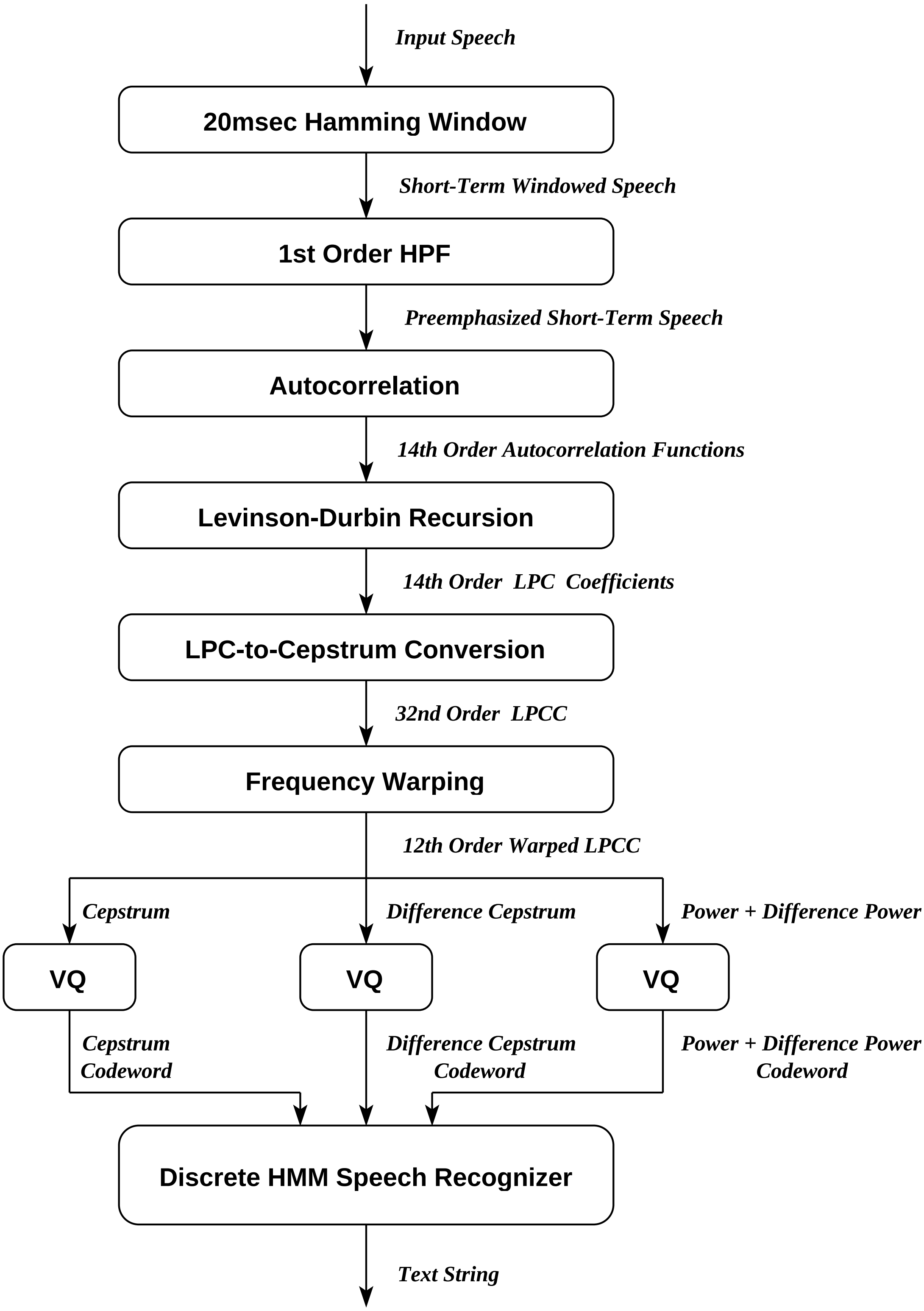
Input Speech
20msec Hamming Window
Short-Term Windowed Speech
1st Order HPF
Preemphasized Short-Term Speech
Autocorrelation
14th Order Autocorrelation Functions
Levinson-Durbin Recursion
14th Order LPC Coefficients
LPC-to-Cepstrum Conversion
32nd Order LPCC
Frequency Warping
12th Order Warped LPCC
Cepstrum
Difference Cepstrum
Power + Difference Power
VQ
VQ
VQ
Cepstrum Codeword
Difference Cepstrum Codeword
Power + Difference Power Codeword
Discrete HMM Speech Recognizer
Text String

Figure 2-1 Block diagram of the 3-codebook discrete HMM SPHINX recognition system.

Similar to ordinary scalar quantization, an error is associated with the process of VQ. This error is called VQ distortion and is given as the distance between the tested feature vector and the chosen centroid. In general, smaller VQ distortion is an indicator that a feature space is well represented by a particular codebook. The number of centroids is also a specification of VQ and is referred as the codebook size. In SPHINX, the codebook size is 256, and three separate codebooks are generated for cepstra, difference cepstra, and energy.

2.3. Discrete Hidden Markov Model

.In SPHINX, the minimal acoustic unit of speech recognition is a special form of a phoneme. The incoming speech to be recognized is now a string of discrete symbols after the VQ. These two are incorporated into a random-process model called the discrete hidden Markov model (HMM)[12]. In HMMs, each model represents a particular phoneme is a network of states emitting output symbols at each state transition forms a building block. It is characterized by two sets of probabilities, the probabilities of state transitions that are conditioned by the identity of the current state, and the output probabilities that give conditional probabilities of output symbols given a state transition.

Thus the task of speech recognition is formulated as one of finding the most likely HMM given an observed symbol sequence. The goal of training the recognition system is to assign proper probabilities for each HMM based on a large enough amount of pooled symbol sequences and their phonetic transcriptions.

In the SPHINX recognition system we use three codebooks. The HMM topology is modified such that each state transition produces a triplet of discrete symbols, each of which are statistically independent.

Chapter 3 Signal Processing Issues in Environmental Robustness

In this chapter, we discuss some of the common sources of degradation of recognition accuracy that a speech recognition system will usually face when noise and distortion are present. We then briefly discuss possible solutions to this problem.

3.1. Sources of Degradation

In general, a robust speech recognition system needs to cope with many factors that cause potential degradations in performance, including variabilities between speaker characteristics and changes in environmental attributes. In the context of environmental robustness in the office environment, the major factors causing degradation of recognition accuracy are additive noise and linear filtering. Other sources of degradation include articulation effects induced by environmental influences (such as the Lombard effect), transient noise with high energy, and interference by speech signals from other speakers talking simultaneously (the cocktail party effect). As it will become clear in the following discussion, additive noise and linear filtering are the two major factors that contribute to degradation of performance of a speech recognition system in our census database in which environmental variabilities come from change between a close-talking microphone and a desktop microphone.

3.1.1 Additive Noise

Performance of speech recognition systems degrades drastically when training and testing are carried out with different noise levels. For example, it was found that when a zero-mean, white Gaussian noise is added to an utterance at various SNRs, the accuracy of speech recognition will drop from 95.6% when clean speech is used, to 46.3% when noisy speech at a global SNR of 15 dB is used[13] [14]. In a passenger car or in the cockpit of a modern jet fighter aircraft noise presents an extremely challenging problem for automatic speech recognition.

In most cases, acoustic ambient noise or background noise is considered to be additive and hence can be modeled as an additive stationary process that is uncorrelated with the speech signal. Many speech enhancement techniques have been proposed to minimize the effect of this corrupting noise, and they have achieved differing levels of improvement in accuracy (e.g. [13][2]).

3.1.2 Linear Filtering

In addition to additive corruption by noise, speech may have undergone a series of spectral distortions while being produced, recorded, and processed for recognition. The room where the speech recognizer is deployed certainly has a varying degree of reverberation that can influence the signal spectrum. The microphone, depending on its type and mounting location, can have significant impact on the speech spectrum. When the microphone configuration used in testing is different from the one used for training the reference patterns, the mismatch in spectral distortion becomes a major problem. Furthermore, when telephone lines are used to convey the speech signal, channel-induced variation can also be regarded as a combination of additive noise and channel filtering. It was also found that the recognition error rate of a speech recognizer could increase from 1.3% to 44.6% when the testing data were filtered by a pole/zero filter modeling a typical long-distance telephone line and corrupted by noise at a global SNR of 15 dB[15].

3.1.3 Other Effects

There are other factors that contribute to the degradation of performance of a speech recognition system. They include articulation effects (Lombard speech), transient noise, and interference from other speakers (cocktail party effect), and so on. These effects are difficult to quantify mathematically. Very few approaches have been proposed to address these effects (with the exception of transient noise). We are not attempting to address these problems in the proposed work, either, as we will focus on the algorithms that address the major sources of degradation in offices: additive noise and spectral tilt by linear filtering.

3.2. Solutions to the Environmental Robustness Problem

In Section 3.1. we identified the major sources for degradation in recognition when mismatches occur between training and testing conditions, as in the case when speech recognizers are confronted with different microphones used for training and testing. In this chapter, we will review signal processing techniques called acoustical pre-processing as an effective solution to the environmental robustness problems.

The application of acoustical pre-processing can be regarded as a signal enhancement technique that attempts to eliminate or reduce the corruption induced by changes in the environment. When the signal degradation is due to additive noise alone, one can utilize enhancement methods to suppress noise before applying the speech recognition algorithm. For example, adaptive noise cancellation[16] uses two signal sources, one a filtered version of the noise, and the other containing the corrupted input (speech plus noise). However, adaptive noise cancellation requires that the signal and distortion be statistically independent, which is not the case for signals in a normal reverberant office environment. Other types of approaches(e.g. [17]) assume the existence of only a single channel containing noise-corrupted speech. Most of these speech enhancement algorithms deal only with additive noise problem, and spectral equalization needs to be performed using some other technique.

Several algorithms that consider the joint effects of additive noise and spectral tilt on corrupted speech signal explicitly have been proposed by Acero[2]. In these algorithms, acoustical compensation consisted of an additive correction in the cepstral domain. Specifically, compensation factors are estimated for additive noise and spectral transformations by minimizing the differences between speech from the training and testing environments using various procedures. Since they operate in the cepstral domain, they are easily incorporated in the SPHINX, which uses cepstral vectors as features. We will describe two major algorithms proposed by Acero in the next subsections.

3.2.1 Review of Algorithms for Cepstral Normalization

In order to model different environmental effects, we use the model shown in Figure 3-1 to describe the two main kinds of degradation that we have discussed in Section 3.1. We assume that the speech signal is passed through an unknown linear filter $h[m]$ whose output is then corrupted by uncorrelated additive noise $n[m]$. Based on this assumption, we characterize the power spectral density (PSD) of the processes involved as:

$$P_y(\omega) = P_x(\omega)|H(\omega)|^2 + P_n(\omega)$$

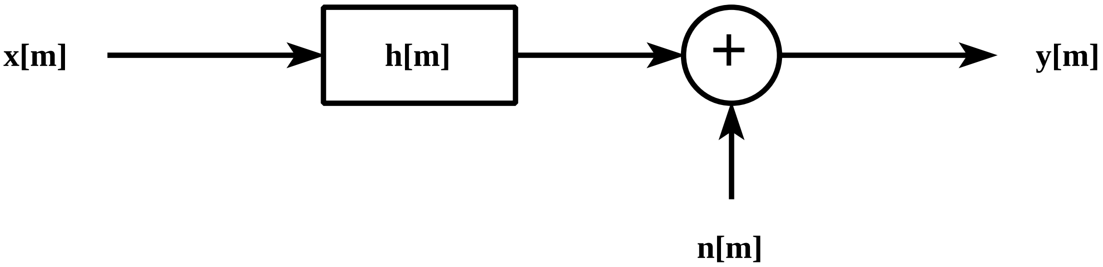

Figure 3-1 Model of signal degradation by linear filtering and additive noise.

If we let the cepstral vectors $\boldsymbol{x}$, $\boldsymbol{n}$, $\boldsymbol{y}$ and $\boldsymbol{q}$ represent the Fourier series expansion of $lnP_x(\omega)$, $lnP_n(\omega)$, $lnP_y(\omega)$ and $H(\omega)$ respectively, the above equation can be rewritten as:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{q} + \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{q})$$

where the correction vector is given by:

$$\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{q}) = IDFT\{ln(1 + e^{DFT\{\boldsymbol{n} - \boldsymbol{q} - \boldsymbol{x}\}})\}$$

We can obtain an estimate of $\hat{P}_y(\omega)$ of the PSD $P_y(\omega)$ from a sample function of the process $y[m]$. If $\boldsymbol{z}$ represents the observation of $\boldsymbol{y}$ (the Fourier expansion of $lnP_y(\omega)$), our goal is to estimate the uncorrupted vectors $\boldsymbol{X} = \boldsymbol{x}_{\mathbf{0}}, \boldsymbol{x}_{\mathbf{1}}, \ldots, \boldsymbol{x}_{\boldsymbol{N}-\mathbf{1}}$ of an utterance given the observation $\boldsymbol{Z} = \boldsymbol{z}_{\mathbf{0}}, \boldsymbol{z}_{\mathbf{1}}, \ldots, \boldsymbol{z}_{\boldsymbol{N}-\mathbf{1}}$.

3.2.2 SNR-Dependent Cepstral Normalization (SDCN)

The first compensation algorithm, SNR-dependent cepstral normalization (SDCN), applies an additive correction in the cepstral domain, with the compensation vector depending exclusively on the instantaneous SNR of the signal. The compensation vectors equal the difference between the average cepstra from simultaneous stereo recordings of speech signals from the training and testing environments for each SNR of speech. At high SNRs, this compensation vector primarily compensates for differences in spectral tilt between the training and the testing environments, while at low SNRs the vector provides a form of noise substraction. This algorithm is simple and effective, but for every new testing environment it must be calibrated with a new stereo database that contains samples of speech simultaneously recorded from the training and testing environments. In many situations, such a database is impractical or unattainable. Furthermore, SDCN is clearly not able to model a non-stationary environment since only long-term averages are used.

3.2.3 Codeword-Dependent Cepstral Normalization (CDCN)

The codeword-dependent cepstral normalization algorithm (CDCN) uses EM (Estimate-Maximize) techniques to compute ML estimates of the environmental parameters that characterize the contributions of additive noise and linear filtering. These coefficients are chosen such that when applied in inverse fashion to the cepstra of an incoming utterance they produce an ensemble of cepstral coefficients that best matches (in the ML sense) the cepstral coefficients of the incoming speech in the testing environment to the locations of VQ codewords in the training environment. Use of the CDCN algorithm improved the recognition accuracy obtained when training on the standard close-talking microphone and testing on the desk-top microphone to the level observed when the system is both trained and tested on the desk-top microphone. The CDCN algorithm has the advantage that it does not require *a priori* knowledge about the testing environment, but it is much more computationally demanding than the SDCN algorithm. While CDCN significantly improves recognition accuracy in cross-microphone situations, it also tends to produce a slight degradation in recognition accuracy compared to the baseline result with no acoustical pre-processing when the system is both trained and tested using the clean speech.

Chapter 4 A Selective Review of Previous Auditory Models

Auditory models are generally designed with the goal of replicating or simulating certain specific aspects of the auditory periphery, based on knowledge of psychoacoustics and/or neurophysiology. The aspects of human audition that have been studied and modelled include the frequency selectivity of the cochlea and the auditory nerve, the saturated half-wave rectification that takes place at the level of the hair cells, and short-term adaptation and automatic gain control (AGC). In this section we review these aspects of auditory processing that have become important components of physiologically-motivated signal processing algorithms for speech recognition.

Speech sounds are converted into mechanical vibration and transmitted to a snail-shaped organ called the cochlea via three small bones of the middle ear. The cochlea is filled with fluid, with which mechanical vibration is converted into fluid motion. Along the turns of the cochlea there are thousands of receptor elements called inner hair cells (IHCs), which are sensitive to fluid motion. Nerve fibers connected to the IHCs convey neural pulses in the form of temporal discharge patterns. The neural activity is stochastic, with the frequency of the pulses increasing as the corresponding hair cells are activated more intensely. In addition, the temporal pattern of nerve firing shows phase-locked characteristics that are synchronized to the fine structure of input stimuli for stimulus frequency components below 1.5 kHz. The cochlea serves as a mechanical frequency analyzer so that each frequency component has its own place of maximum resonance along its length. Groups of hair cells in a localized region along the cochlea pass on information to the fibers of the auditory nerve, and each auditory-nerve fiber is maximally responsive to a specific frequency referred to as the characteristic frequency (CF) of the fiber. It is less well understood how information is processed at higher, more central, levels. Nevertheless, the auditory-nerve fibers are the only major outgoing information path from the auditory periphery to the brain, so information that auditory-nerve fibers collectively convey should be very useful in constructing time-spectral images of sound.

Our hope has been to be able to identify the key components which play important roles in extracting coherent spectral information of the original signal out of the degraded input. Figure 4-1 describes a common functional organization that is used in most models of auditory processing. As shown in Figure 4-1, the auditory periphery can be characterized by three cascaded stages of different generic types of signal processing:

- A linear bandpass filter bank to model the frequency selectivity of the cochlea
- Non-linear transformations to simulate physiological phenomena observed at the neural transduction (NT) level
- Higher-level processing to extract coherent spectral images and other important information

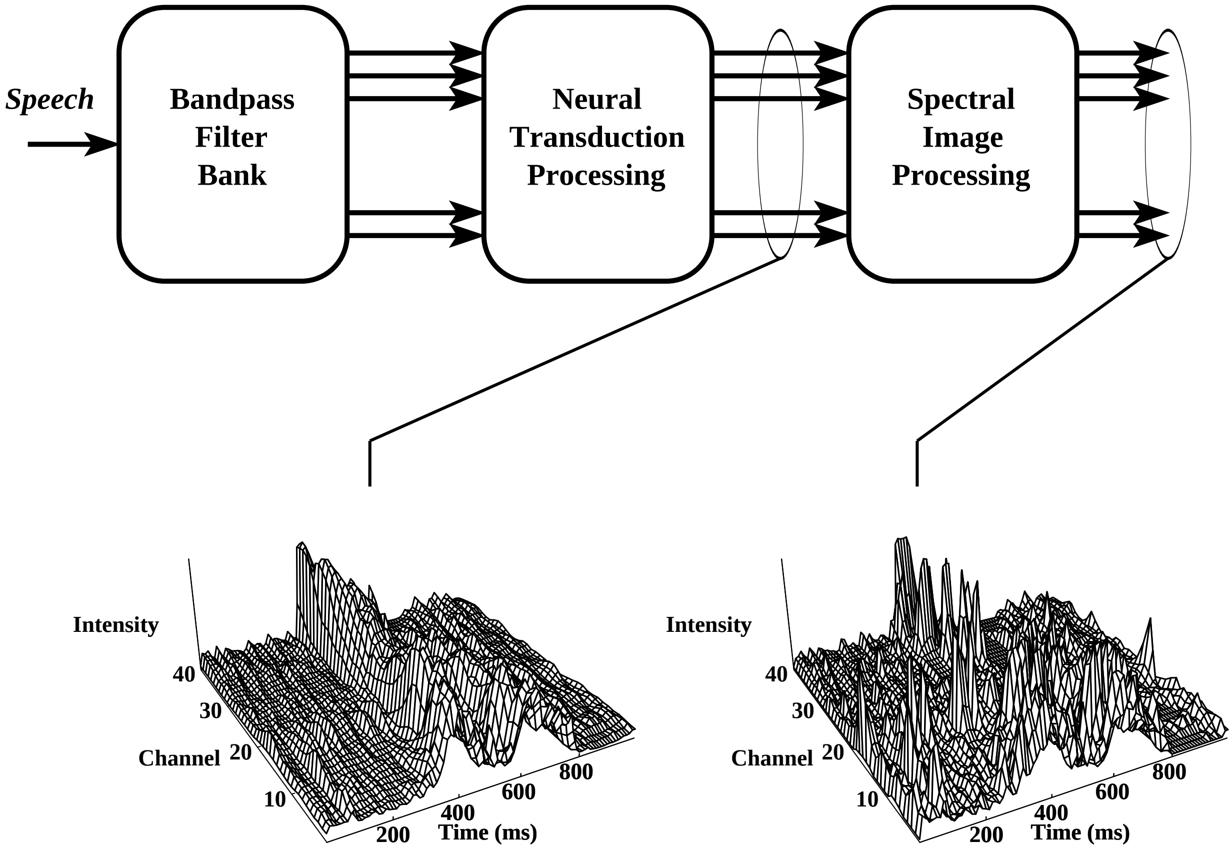

Figure 4-1 Block diagram of physiologically-motivated signal processing. The input speech is passed through the BPF that simulates frequency selectivity of the cochlea, then processed by modules that simulate nonlinearity at the neural transduction level to produce short-term energy spectrum of speech. Finally, higher level processing of channel signals provides alternative representations of speech.

4.1. The Bandpass Filter Bank

The purpose of the bandpass filter bank (BPF) is to simulate the frequency-selective signal processing that is observed at the level of the cochlea and the auditory nerve. We discuss in this section the major approaches used in designing BPFs, and other related issues.

Most auditory models include a set of BPFs that mimic the place-dependent tuning characteristics of the cochlea and the connected auditory-nerve fibers. While there is considerable variation in the specific design procedures used by the various researchers, some common principles are seen in all designs.[18][19]. For example, each filter has a steep high-frequency roll-off above resonance with a slope of about -120dB/octave, so frequency components of an incoming signal above the resonant frequency for a given channel are sharply attenuated. The low-frequency skirts of the filters are not nearly as sharp as the high-frequency skirts. In addition, channel bandwidth becomes broader as center frequency increases, so the filters in the filter bank provide finer frequency resolution in the low-frequency channels, but better time resolution in the higher-frequency channels. These design principles are motivated by extensive physiological measurements of the auditory-nerve response to simple stimuli (*e.g.* [20]).

The asymmetric filter shape and changing filter bandwidth is seen in the shape of the filters used in the Seneff model [7]shown in Figure 4-2 plotted in the linear frequency and in Bark scale[21] The bandwidth of the channel filter element increases exponentially above 1.2kHz reflecting the nature of human auditory periphery. In Seneff model, the following formula proposed by Goldhor[22] was used:

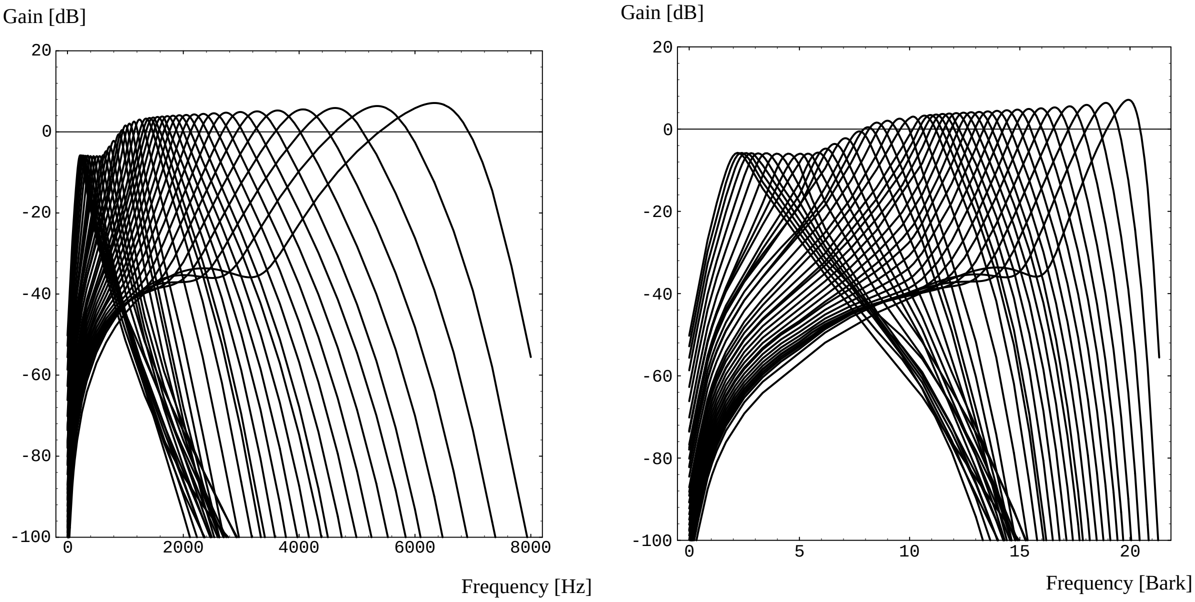

Figure 4-2 Magnitude response of Seneff BPF plotted in the linear frequency and in the Bark scale.

$$B(f) = \begin{cases} 0.01f & 0 \le f < 500 \\ 0.007f + 1.5 & 500 < f < 1220 \\ 6ln(f) - 32.6 & 1220 < f \end{cases}$$

Also there is another nonlinear frequency scaling called mel frequency scale [23], which is also widely used:

$$M(f) = \frac{1000}{\log 2}\log\left(1 + \frac{f}{1000}\right)$$

In addition to designing the magnitude response of the BPF based on the tuning curves of the cochlea and the auditory nerve fibers, Seneff[19] tried to replicate the phase response of the cochlea, by using extra poles and zeroes. In particular, Seneff was concerned with obtaining the physiologically-correct phase alignment among channels because she subsequently made use of the summed channel outputs for pitch estimation. However, it is possible that the phase characteristics if the BPF may not be as important as the magnitude characteristics for our purpose, which is to obtain an empirically-useful short-term frequency representation of speech.

In general, it is possible that the bandpass filters need not reflect the exact nature of the narrowband tuning characteristics of the cochlea and auditory nerve fibers if one were not concerned with building a spectral image by combining individual channel information properly. For example, Ghitza[24] initially formulated a model which used BPFs that approximated the characteristics of the cochlea, but in later experiments he obtained equally-accurate speech recognition by replacing his physiologically-motivated bandpass filter shapes by equally-spaced Hamming-shaped BPFs[5]. His results showed that the use of Hamming-shaped BPFs produced better recognition performance. This illustrates that the employment of an auditory processor may have more significance than the choice of different front-end BPFs.

4.2. Neural Transduction (NT) Functionalities

We expect that some of the nonlinearities seen at the level of the IHCs in the auditory periphery would help enhance the time-spectral pattern of speech, and that the transformed information has some of the desirable properties for speech perception. In the time domain, the adaptive nature of the auditory nerve firing reduces the effect of noise if it is stationary, or if it is slowly changing with respect to the transient part of speech. The amplitude of the original signal is compressed to negotiate the relatively narrow dynamic range of the biological system.

A computational model could be constructed by replicating these characteristics, and a well-designed one would simulate some aspects of the human system. The block diagram of Seneff model NT signal processing is shown in Figure 4-3 . However, what is important for us to understand is how the characteristics of individual components of the processing benefit human speech recognition. For example, the compression of signal amplitude may be a consequence of the limitations of the physiological “hardware” and may not be necessary for automatic speech recogni-

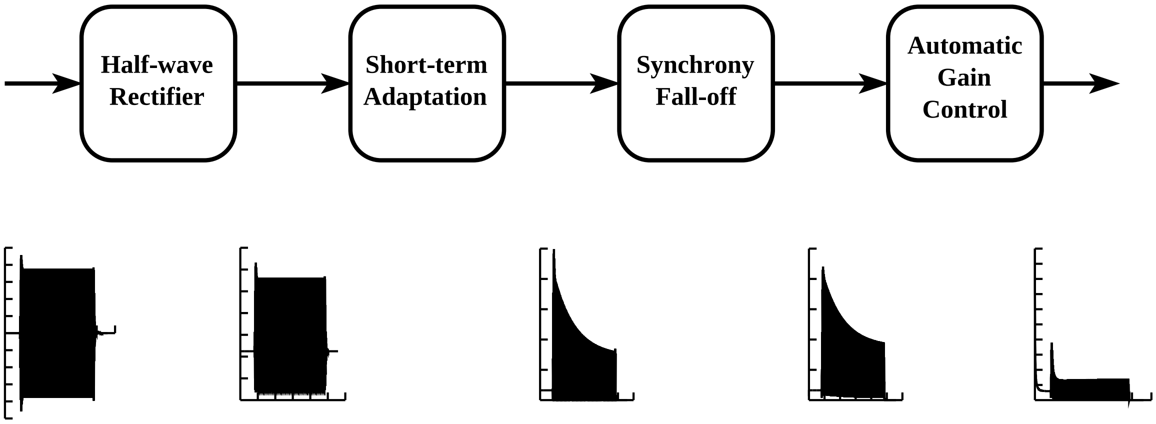

Figure 4-3 Block diagram of NT processing by the Seneff model. Also plotted are the inputs and outputs at each NT module past the BPF. The input signal is a 100-msec tone burst of 2 kHz, and the response is observed at the channel with CF equal to 2 kHz. Plot axes are arbitrarily scaled.

tion. On the other hand, we believe that the time-varying and adaptive nature of the system would be very useful for speech recognition. It enhances the contrast between the rapidly-changing parts and stationary or slowly-changing parts of the input signal. This distinction is potentially valuable, as rapidly changing frequency components commonly result from transitional segments of speech such as stops, while more stationary segments might correspond to other types of speech sounds such as static vowels.

In the following sections, we discuss the function of individual modules in previously-proposed auditory models, and we attempt to identify ways in which this processing might be useful for our purposes.

4.2.1 Half-wave Rectification

Halfwave rectification is used to characterize the fact that the IHCs are only sensitive to fluid motion in one direction. In various models integration of the output of the half-wave rectifier provides a measure of average energy.

The models of Seneff[19][7] and Shamma[25] also include instantaneous amplitude compression characteristics at the level of the rectifier, based on physiological knowledge of the narrow dynamic range of the auditory system, and its saturation in response to loud stimuli. While it is not clear whether this nonlinear compression is necessary for automatic speech recognition, the nonlinearity does help to recover the rapid onset of tone stimuli that had been lost by passing signals through the narrowband channels of the BPF.

4.2.2 Short-term Adaptation

The firing rate of an auditory-nerve fiber is highest during the initial 15 ms of the stimuli and decreases to a steady-state level after 50 ms or so. (The peak firing rate of the fiber is about 800/s.) This process is known as adaptation. Because of adaptation, the auditory periphery remains responsive to the time-varying part of the amplitude envelope of the input stimuli but becomes less sensitive to the segments during which the amplitude remains in more of a steady state.

Adaptation and the effects of the AGC are expected to affect the dynamics of the amplitude envelope of the channel signal, although their effect on the final output may not always be significant.

4.2.3 Synchrony Fall-off

The probabilistic behavior of the auditory-nerve firing tends to synchronize to the fine structure of the input in the low-frequency range. Above 1.5 kHz, this phase locking decreases and eventually becomes extinct, a process known as synchrony falloff. Although high-CF fibers do not phase-lock to the input stimuli in their best frequency ranges, their discharge rates appear to be synchronized to low-frequency fluctuations in the envelope of the input, if such fluctuations are present in the signal. Seneff modelled the effect of synchrony falloff by inserting a lowpass filter (LPF) between the adaptation stage and the AGC, although this aspect of processing is not included in the models developed by other researchers. It is not clear what role (if any) synchrony falloff plays in processing speech signals monaurally for robust speech recognition, although it is definitely an important attribute of binaural signal analysis.

4.2.4 Automatic Gain Control

While automatic gain control (AGC) mechanisms are included in various different auditory models, the reason for their inclusion can vary from case to case. For example, Seneff used an AGC mechanism (in conjunction with short-term adaptation) to model the effect of refractory behavior of the auditory nerve, which “locks out” subsequent firings for about 2 ms after each pulse. Lyon, on the other hand was primarily interested in the use of AGC mechanisms with differing time constants (10ms, 40ms, 160ms, and 640ms in his model) to obtain the general capability of gain control and inhibitory interactions among adjacent channels.

4.2.5 Lateral Suppression

Another interesting characteristic of the auditory-nerve fibers is lateral suppression, which is a reduction of the response of a fiber with a given CF by stimulus components at adjacent frequencies. The amount of suppression depends on the general spectral profile of the stimulus, and specifically on the relative energy of stimulus components at and adjacent to the CF of the fiber. Lateral suppression produces enhanced local contrast in the spectral profile of the response to a stimulus, which may be useful in reconstructing the spectral image. In general, the inclusion of suppression into a model of the auditory periphery will help emphasize the peaks of less obvious spectral images, but it may also amplify the effects of additive noise in the spectral representation,.

Lyon[26] and Shamma[25] both included mechanisms to model the effects of lateral suppression in their models. In Lyon’s model, lateral suppression is realized using AGC mechanisms. The outputs of three adjacent channels are combined after the other nonlinear processing stage, with the center channel providing excitatory input and the two adjacent channels providing inhibition. Shamma implemented a suppression mechanism with two networks. The first network is prior to the half-wave rectifier, and consists of a simple linear combination of seven adjacent channels with symmetric weights of 0.02, 0.08, -0.3,0.4, -0.3, 0.08, and 0.02. The second network is a single layer of neurons with negative feedback interconnections, from which the steady-state output is computed by iteration. The purpose of the secondary network is to sharpen the output of the first network, which realizes the inhibitory characteristics. Although the overall effect of the two lateral suppression networks is not straightforward to analyze, the major effect of the first network is to reflect the status of two neighboring channels.

4.3. Spectral Images Based on Synchrony Information

The goal of the models discussed up until now has been to characterize the response of the auditory system at the level of the auditory nerve to simple stimuli. It is also not presently well understood how the central auditory system interprets the information transmitted by the auditory-nerve fibers. Since at present it is difficult to characterize higher-level processing solely on the basis of the limited physiological data obtained thus far, several researchers have proposed various computational ways of interpreting auditory-nerve information that would enhance perceptually-significant features such as formants, fundamental frequency, and timbre. These central-processing models are (for the most part) motivated to a greater extent by characterizing function as opposed to physiology.

In this section we review some of the ways that have been proposed to construct robust spectral images from the information that results from hypothesized auditory processing beyond the obvious use of mean rate of auditory-nerve response as a function of the CF of each fiber. Most of these processing schemes exploit the synchronous response of auditory-nerve fibers to the fine structure of low-frequency components of sound. We first review the work that motivated researchers to observe synchrony information as a consistent measure that preserves spectral information of the original signal. We then look at several proposed signal processing algorithms and discuss their similarities and differences in terms of synchrony detection.

In the late 1970's, Young and Sachs demonstrated that the use of temporal information observed in nerve firing patterns helps produce more consistent spectral images than merely the use of mean-rate information. Specifically they proposed a measure called Average Localized Synchronized Rate (ALSR)[27]. The ALSR is obtained from interval histograms of auditory-nerve firing patterns by computing the normalized intensity of the phase-locked component of response at the CF of each fiber and the by averaging it locally within 0.25 octaves. A measure of relative spectral prominence is obtained from local averages of these phase-locked components across fibers with similar CFs. The ALSR was found to be very useful in constructing robust spectral images that resemble the original input speech in the presence of noise. These responses retain information pertaining to the major resonant frequencies of speech sounds over a wider dynamic range than the information provided by spectral estimates obtained from mean-rate information. In other words, analysis of physiological data showed that the normalized synchrony measure preserves information related to prominent spectral features such as formant peaks in the presence of noise, and in a way that is less subject to input amplitude variation.

In the sections below we have a selective review of three models that have attempted to extract and interpret this synchrony information by reinforcing the periodicity of the waveform. The scope of the review is limited and by no means comprehensive. There are other models of spectral image processing such as Ghitza's ensemble interval histogram (EIH)[28] [29]that are also important but not covered due to little relevance to our work.

4.3.1 Synchrony and Mean-rate Spectra

As described in the previous section, the channel output signals from the model for auditory processing are intended to approximate the probabilistic firing behavior of auditory-nerve fibers. Since channel elements have different CFs, the average outputs of these channels viewed as a function of frequency provide a spectral image, which Seneff, for example, refers to as the mean-rate spectrum. However, the mean-rate response by itself may not be the best display for speech recognition, as Sachs and Young demonstrated experimentally. Although it does provide robust cues for broad categorization of gross spectral features, the mean-rate spectrogram looks less distinct than the conventional spectrogram based upon Fourier analysis in representing the fine structure of the actual spectrum of the speech sounds. One problem with the mean-rate spectrum, for example, is that individual channels can saturate in their response to intense stimulus components below CF, which can produce a saturated spectral image for many CFs for many stimuli.

Motivated by the physiological findings of Sachs and Young, Seneff proposed a simple detection mechanism called the Generalized Synchrony Detector (GSD) which responds to periodic waveforms by comparing the similarity of the input signal to a delayed version of itself. The GSD measure $y(t)$ as function of the channel signal $x(t)$ is given by:

$$y(t) = Af\left(\frac{1}{A}\frac{\langle|x(t)+x(t-\tau)|\rangle-\delta}{\langle|x(t)-x(t-\tau)|\rangle}\right)$$

$$\tau = CF^{-1}$$

where τ is the delay time set equal to the inverse of each channel's CF, δ is a constant to correct the d.c. offset associated with the modelling of spontaneous firing rate. $\langle\bullet\rangle$ denotes the averaging operation by leaky integration. *f*( ) is a soft compression function and is typically $\tan^{-1}(\ )$. A is a constant to adjust linearity and output range. The GSD produces high output values when the signal has the periodicity with the period τ that results in the zero denominator value. The GSD is somewhat similar to the average magnitude difference function (AMDF)[30] [31]:, which approaches

$$D_k = \frac{1}{N}\sum_n|X[n]-X[n-k]|$$

zero when $X[n]$ is periodic with the period of k, providing a more efficient indicator of periodicity than the autocorrelation functions:

$$R[k] = \sum_n (X[n]X[n-k])$$

And like the AMDF, the GSD responds to stimulus components at frequencies that are integer multiples of the CF as well as the CF itself (for τ=1/CF). Hence, the BPF applied prior to the GSD must steeply attenuate input frequency components above the CF if the GSD mechanism is to respond only to the stimulus component at frequency CF. By cascading the model for peripheral processing with the GSD, the amount of synchronization of each channel signal can be detected in a way that is more robust to additive noise and changes in input level (according to the results of Sachs and Young).**]**

In developing the GSD, Seneff also considered versions of the short-term autocorrelation function, but she found that an algorithm based on the AMDF provided sharper synchrony-detection capability. The definition of the GSD is somewhat *ad hoc,* but its ability to enhance spectral peaks more sharply than the autocorrelation function has been confirmed for speech-like stimuli.

Because of their different characteristics, the combined use of mean-rate and synchrony spectra would be expected to provide a more informative set of features. Meng[6] compared the recognition accuracies obtained in a vowel-identification task using the mean-rate and GSD representations separately and in combination. To summarize Meng's results, both the mean-rate and synchrony mechanisms provided substantial advantage over the level of performance obtained using conventional signal processing based on linear predictions, and the optimal combination of mean rate and synchrony (obtained using a principal-components analysis) performed the best. These differences were true for both clean and noisy speech data, and is particularly dramatic with noisy test data.

Since Meng's task was limited to the recognition of vowels, this study can not be sued to confirm the anticipated utility of mean-rate representations for detecting weak events in aperiodic signals. Meng used a neural-network-based pattern classification scheme to perform her evaluations, and it is not clear if her good results with physiologically-motivated models will also hold true for the more prosaic classifiers that assume multivariate normal distributions, as the classifiers used in SPHINX. The performance improvement obtained when the two representations are used in consort may indicate that there is something missing in both the mean-rate and the GSD parameters that becomes useful when they are combined.

4.3.2 Cochleagrams and Correlograms

The cochleagram and correlogram displays were proposed by Lyon[32] as graphic ways to represent the output of his model. Although they both resemble speech spectrograms, they are different in motivation and in terms of information content. The cochleagram is a direct visualization of the channel outputs from the AGC and is therefore similar to Seneff's mean-rate spectrum. The correlogram, as its name implies, is the visualization of the running autocorrelation functions of the outputs of each of the bandpass channels and it is meant to provide an alternative tool to model pitch perception[33]. (The correlogram, which has also been called the correlogram, was originally intended to model binaural localization by computing the cross-correlation functions of the outputs of channels with matched CFs from each of the two ears[32].)

In the correlogram, the fundamental pitch period of a speech segment is identified by looking for a strong dark line close to the origin that runs vertically across frequency at the delay time position. An “edge” filter is used to enhance contrast along the delay-lag axis for better resolution.

Like the GSD, the correlogram is clearly based on the autocorrelation of the incoming signal after peripheral bandpass filtering. However, since the correlogram is a function of two variables (CF and internal delay), it (in principle) provides more information than the GSD processor. In fact, if we consider a correlogram plot, the information provided by the GSD processor falls along the single curve in the delay-time-CF plane for which delay is the reciprocal of CF.

4.3.3 Triggered Quantized Time Integration

The Triggered Quantized Time Integration (TQTI) technique was proposed by Patterson[34] as a model that handles the perception of the pitch, phase, and timbre in a unified fashion. Its spectrographic presentation gives somewhat similar images to Lyon’s correlogram. The major difference between the two models is that TQTI makes use of local amplitude maxima of the outputs of the bandpass channels as timing information for segmentation and time alignment as explained below.

The underlying idea of TQTI is similar to pitch-synchronous analysis. The output signals of each bandpass channel are marked by the times of maximum amplitude, and consecutive short segments of these signals are time aligned according to these maximum-amplitude marks and summed up to form an averaged waveform. The maximum amplitude timings in the input waveform would be closely related to the pitch excitation. On the other hand, the noise component is not coherent to such timing information. Averaging several time-aligned waveform segments would enhance the pitch synchronous components of the signal obscured by the effect of noise.

It is not clear whether TQTI would always provide us with more useful information than the GSD or the correlogram for the purpose of constructing robust spectral images over a wide range of SNRs. Using the timing information could either be a merit or a drawback. In high-SNR situations, triggering guarantees that the waveform segments are time aligned and summed to enhance their common features without sacrificing fine temporal information, which may be beneficial. On the other hand, when the SNR is low, coherent timing information is no longer available and we may have to encounter a threshold beyond which TQTI processing will not work as the SNR decreases.

4.4. Summary

In the previous sections, we have reviewed three major functional blocks commonly found in various models of auditory periphery, the bandpass filter bank, the NT signal processing, and the extraction of synchrony information. The frequency selectivity of the cochlea is characterized by the asymmetrical filter shape and the non-uniform spacing and bandwidth of channel filter elements. The NT signal processing includes several nonlinear functions to model relevant physiological phenomena. Finally, we have reviewed three models for building spectral image by use of periodicity of the signal.

4.5. Analysis of Computational Complexity of Some Major Auditory Models

Table 4-1 provides comparisons of the three major auditory models by Seneff, Lyon, and Ghitza respectively[7][26][29] in terms of computational complexity. These numbers can also be compared to the computational complexity of conventional front-ends based on LPC techniques. In the SPHINX system, for example, the total computational cost for the front end is about 600 multiplies and adds to process 1 msec speech segment. In the comparison, a non-optimized implementation of Seneff's model runs 22 times real time, whereas the SPHINX front end runs in real time on the DEC5000/200.

As is discussed later in Chapter 6, from the execution profile of Seneff model we have learned that the BPF takes about 20% of the whole computational cost. Lyon's BPF is about twice as efficient as Seneff's is considering that the number of channels is 85 as opposed 40 by Seneff, which implies that the additional cost for modelling the phase characteristics of the cochlea is not trivial. In Seneff's model, the last stage in computing the GSD response is the most demanding part computationally; therefore, any substantial improvement in efficiency at the GSD processor would have a great impact on the entire system. We have found that the processing takes up about 37% ~ 56% of the CPU cycles depending on the degree of optimization of the whole module.

Model		Seneff	Lyon	Ghitza	SPHINX
Number of Channels		40	85	85	14[a]
Sampling Frequency		16kHz	16	6.67/40[b]	16
BPF	Mults/ms	8448	7442	110k	N/A
	Adds/ms	7808	6832	177k	N/A
NT	Mults/ms	8000	16320	N/A	N/A
	Adds/ms	5120	24480	N/A	N/A
	Func/ms[c]	640	0	N/A	N/A
Spectral Image	Mults/ms	7040	27200[d]	0	N/A
	Adds/ms	6400	54400	429k[e]	N/A
	Func/ms	1280	0	0	N/A
Total	Mults/ms	23488	50962	110k	600
	Adds/ms	19328	85712	606k	580
	Func/ms	1920	0	0	0

a. LPC analysis order
b. 6x oversampling
c. C math library calls such as atan ( )
d. Correlogram computed up to 20 msec lag
e. 191k of logical comparisons are included here as adds

Table 4-1 Comparison of auditory models in computational complexity. Tabulated are numbers of multiplies, adds, and C-math library calls that are necessary to compute for all channel outputs in order to process 1 msec duration of speech. Numbers for the Lyon and Ghitza models are estimated based on a study of their algorithms and not from actual implementation.

Chapter 5 Baseline Recognition Performance of Physiologically-Motivated Front-Ends

5.1. Introduction

In this chapter, we investigate the performance of physiologically-motivated front-ends in various settings using the SPHINX speech recognition system, and we compare these data with similar results obtained using the conventional LPCC front end. These results provide a baseline level of performance to which later results will be compared.

We first describe the experimental paradigm with which the experimental work was developed, including software and data resources, as well as the ways in which the experimental outcomes are analyzed. We then compare the recognition performance obtained using the SPHINX speech recognition system in conjunction with several front-end signal processing algorithms including both conventional linear prediction and physiologically-motivated signal processing algorithms.

5.2. Software and Data Resources for Experiments

We used a modified version of CMU's SPHINX speech recognition system. The conventional LPC-based front end for the SPHINX system as described in Chapter 2 was replaced by a candidate physiologically-motivated front-end, which included several components to represent the BPF bank, linear and non-linear half-wave rectifiers, short-term adaptation and rapid AGC stages, synchrony fall-off, and synchrony detection as described in other chapters of the thesis. There was also the instantaneous compression of the amplitude associated with the use of a non-linear half-wave rectifier. Our baseline model was based on the model of Seneff, because this model is the most completely described in the open literature, and is also available in source-code form. The rest of the SPHINX system remained intact, except for changes in the interface to accommodate the modifications of the front end. At the present time, recognition experiments using versions of the modified front end are still quite expensive, and the turnaround time for each train-and-test cycle is about 40 hours.

For training and testing the modified SPHINX system we used speech data as described in the following subsections, depending upon which specific issues we focused on for each experiment.

5.2.1 The AN4 Stereo-recorded Speech Database

The speech database, which for historical reasons is referred to as the "AN4" database, consists of two disjoint sets of utterances: 1018 training utterances collected from 53 male and 21 female speakers, and 140 testing utterances from 7 male and 3 female speakers. The total length of the training utterances is about 2800 seconds. Different speakers are used for the training and testing data.

The utterances consist of letters of the alphabet, digits, more complex numbers, some control words (such as “yes”, “no”, “enter”, and “repeat”), and strings of letters taken from census data such as name, address, and telephone number.

Both the training and the testing utterances in the alphanumeric database were recorded simultaneously in stereo using two different microphones, (1) a Sennheiser HMD-224 close-talking microphone attached to the headset worn by the speakers, and (2) a Crown PZM6FS omnidirectional microphone placed on the desk top nearby. We use speech data obtained using the close-talking microphone (CLSTK) as the model for clean data in training the system, as well as for tests to provide a baseline level of performance. We use data obtained from the omnidirectional PZM microphone (CRPZM) to provide a real example of speech that contains the combined effect of reverberation and additive environmental noise. The omnidirectional sensitivity and the distant placement of the CRPZM microphone introduces such degradations. By using the CRPZM data for the testing, we are able to evaluate the robustness of the proposed system for practical use.

5.2.2 Speech Data with Simulated Degradation.

Although one of the major goals of this research is to develop an environmentally-robust speech recognition system in realistic practical environments, we also need to be able to evaluate the functioning of individual processing modules in a controlled fashion. Hence we would to be able to separately introduce the effects of reverberation and noise.

The effects of additive noise were simulated by adding white Gaussian noise to the CLSTK data at various segmental SNR levels. The only effects of room acoustics considered were the natural degradations introduced by use of the CRPZM data. The analysis of reverberation by itself is a very difficult problem and is not isolated for deconvolution of the effects of linear filtering sense[35], and there is no easy way to simulate the effects of room acoustics in a realistic but controlled manner.

5.3. Testing of Statistical Significance

The most important figure of merit for the systems and signal processing schemes considered is the recognition accuracy obtained for the AN4 task. Recognition accuracy is calculated by comparing the string of words recognized to the string of words that had been uttered using a standard nonlinear string-matching program, and by subtracting the percentage of errors caused by insertion, deletion and substitution of words[36][37].

It is important to be able to determine the statistical significance of small differences in recognition accuracy among similar systems in order to interpret experimental results in an objective fashion. Gillick and Cox[38] proposed use of McNemar’s test and a matched-pairs test, in which object tokens--sentences and sentence segments respectively--are treated as statistically independent. Marcus proposed use of several tests including the signed-rank test and the *t*-test, which were found to be more reliable than McNemar's test and to be useful, in particular, when samples size was small[39]. NIST has recently developed a series of automated benchmark scoring programs to evaluate statistical significance, which includes the proposed techniques. [40]

5.4. General Results Obtained Using the LPCC and Seneff Front-Ends

For the remainder of the chapter, we will compare the recognition performance obtained using the SPHINX speech recognition system in conjunction with several front-end signal processing algorithms, including both conventional procedures based on linear prediction and physiologically-motivated processing schemes.

In this section we compare the baseline performance obtained using the LPCC and Seneff model for front-end signal processing. We selected the LPCC as the most representative type of conventional signal processing widely used in speech analysis and recognition (and also because it was the type of signal processing in use in the standard SPHINX system when this work began, and we chose the Seneff model as representative of signal processing techniques motivated by the study of auditory physiology.

Figure 5-1 compares recognition accuracy obtained using the SPHINX system using the LPCC front end and using both the mean-rate and GSD outputs of Seneff's auditory front end, evaluated on the AN4 task. The conditions evaluated include all four permutations of training and testing using the CLSTK microphone (which represents the standard "clean" recording conditions) and the CRPZM microphone (which represents somewhat degraded conditions as encountered in an open-plan office using a desktop microphone.

Front-End	CLSTK Training		CRPZM Training	
	CLSTK Testing	CRPZM Testing	CRPZM Testing	CLSTK Testing
LPCC	85.3%	18.6	76.5	36.9
Mean-Rate	80.0	32.3	69.2	50.1
GSD	75.7	48.7	62.7	53.0

Table 5-1 AN4 word recognition rates obtained by use of features consisting of LPC cepstrum coefficients (LPCC) and the mean-rate and GSD outputs of Seneff's model. Training and testing were done using both the primary close-talking microphone (CLSTK) and the secondary omnidirectional desktop microphone (CRPZM). A language weight of 4.0 was used for the LPCC, and 7.0 for the outputs of the Seneff model.

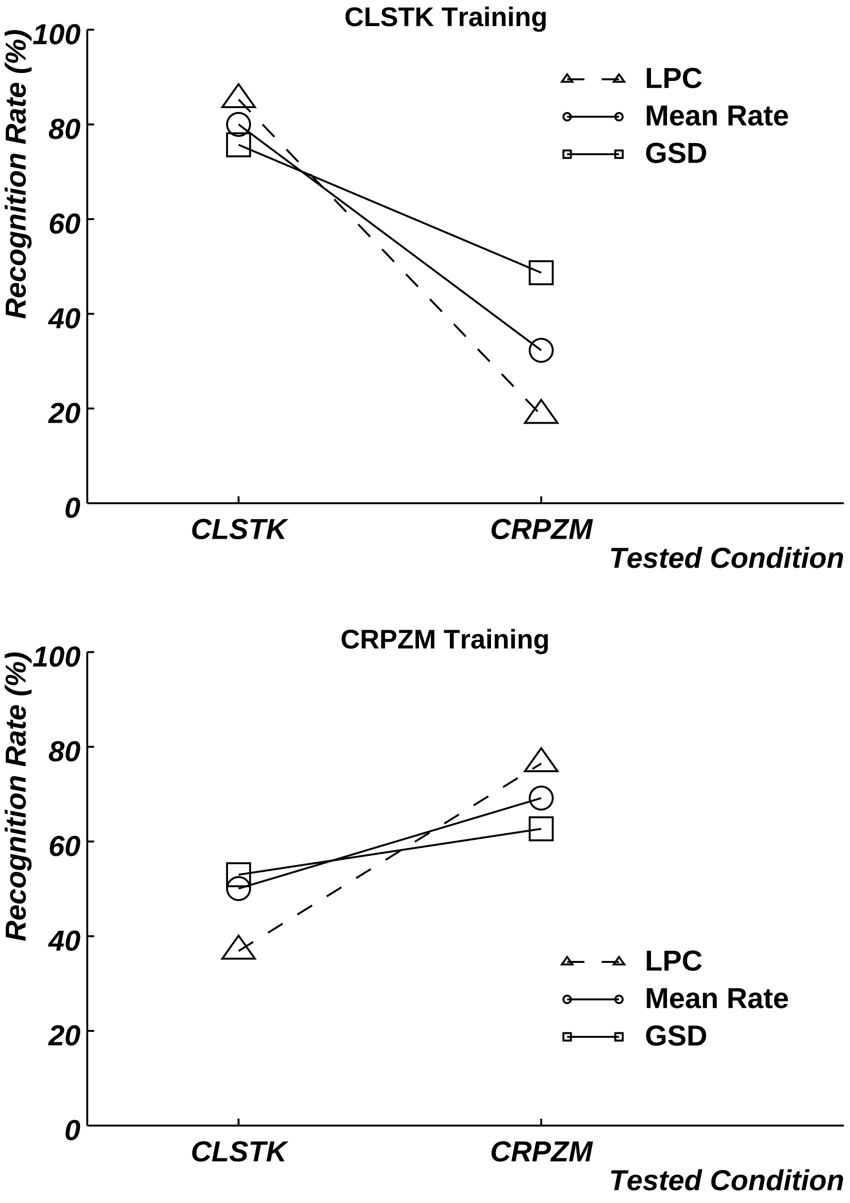

Figure 5-1 AN4 word recognition rates obtained by use of features consisting of LPC cepstrum coefficients (LPCC) and the mean-rate and GSD outputs of Seneff's model. Training and testing were done using both the primary close-talking microphone (CLSTK) and the secondary omnidirectional desktop microphone (CRPZM). A language weight of 4.0 was used for the LPCC, and 7.0 for the outputs of the Seneff model

It is clear that when the same microphone is used for training and testing, the LPCC front end performs the best, followed by the Seneff mean-rate output and GSD output, in that order. The figure of 85.3% obtained using the CLSTK microphone for both training and testing combined with LPCC signal processing is currently the best performance available using the present database and speech recognizer. When a different microphone is used for training and testing, the order of relative performance reverses, with best results obtained using GSD processing, followed by mean-rate processing from the Seneff model and LPCC, in that order. The recognition accuracy of 48.7% obtained using GSD processing was the best result obtained in the important “cross” condition of training using the CLSTK microphone and testing using the CRPZM.

In the CRPZM testing, though, the Seneff model parameters showed obvious advantage over the LPCC, and the order reversed The GSD was the best performing module among the three. We also have found that the GSD is the best front-end for the CRPZM testing among all the front-end algorithms investigated in their raw feature parameter form. Thus, we chose 48.7% by the GSD to be the baseline of the CRPZM testing.

The mean-rate output of the Seneff model performed reasonably well, and better than originally expected. Visual inspection of time-spectral representation of the mean-rate output of the Seneff model reveals that the shape of spectral envelope is very dull, showing a strong compressive nature within a narrow dynamic range. While the mean-rate spectra appear less informative to casual human visual inspection, compared with the outputs of GSD and with LPCC-derived spectral representations. We note that it is frequently misleading to judge the usefulness of feature parameters by visual inspection of spectra.

Finally, we note that when training and testing microphones are different, training using the more degraded environment results in better performance. Nevertheless, in interpreting these results we should bear in mind that the language weight for the recognizer was optimized for performance while training and testing using the CLSTLK microphone. As we shall be discuss in Section 5.7., the best choice of language weight changes if the training and/or testing environments are changed, and there is a distinct possibility that better results might have been obtained using a different language weight for the more difficult CRPZM speech.

Though our focus is on results obtained using the CLSTK microphone for training, it is interesting to observe that the degradation experienced when a different microphone is used for testing is reduced when the system is trained using noisier speech than is afforded by CLSTK training.

5.5. Other Physiologically-Motivated Front-Ends

In this section we briefly review the performance obtained using several other types of signal processing schemes that are less computationally expensive than front ends like those of Seneff, Lyon, or Ghitza, but that are to a greater or lesser degree more similar to physiological processing than conventional liner prediction. As is briefly described in Chapter 4, the use of non-uniform frequency scaling such as mel-frequency scale and Bark scale is based on the non-uniform frequency analysis of the auditory periphery. Popular DFT-based energy spectrum realization based on these concepts include mel-frequency spectral coefficients (MFSC) and Bark auditory spectral coefficients (BASC). The derived cepstral representations obtained after having applied the discrete cosine transform to the MFSC and BASC are the mel-frequency cepstral coefficients(MFCC)[41] and Bark auditory cepstral coefficients (BACC)[42], respectively. Hermansky’s perceptual linear predictive (PLP)[43] analysis also starts with a DFT-based auditory energy spectrum of speech, but it incorporates another aspect of auditory processing, a cubic-root intensity-loudness compression. The magnitude of each DFT bin is compressed by using a cubic-root function to approximate the perceived loudness level. The PLP processing also applies a weighting function to the derived cepstral coefficients so that the resulting distance measure is optimized for a particular task:

The weighted cepstral distance d_{wcep} is defined by

$$d_{wcep} = \sum_i (w_i (c_{test_i} - c_{ref_i}))^2$$

where $w_i$is the weight for the i^{th}order coefficient defined by:

$$w_i = i^S \qquad S \geq 0$$

Note that $S = 0$ gives the Euclidean distance. The optimal value for speaker-independent recognition is found to be $S = 0.4 \sim 0.6$ [44]

We compared results obtained using the MFCC, BACC and PLP signal processing approaches and several different types of feature vectors. For MFCC, we first used the default 12th-order MFCC derived from the putative outputs of a bank of 40 bandpass “filters” which were computed by triangularly weighting the magnitudes of 512-point DFT coefficients. The “filters” have linearly-spaced center frequencies from 200 Hz up to 1 kHz for the lower 12 channels and logarithmically-spaced center frequencies with a channel spacing ratio of 1.07117 up to 6.4 kHz for the remaining channels. We compared filter spacing of the mel-frequency BPF with that of Seneff’s BPF and noted that the former was more sparse than the latter at low frequencies. We were interested in checking the effect of a finer channel spacing. Thus we considered a 12th-order MFCC representation based on 53 bandpass filter channels with a channel spacing that was reduced by 25%. We also tried a 16th-order MFCC based on the default BPF specifications. The motivation for this modification was the pilot experiment by Hwang[45], in which a marginal improvement was observed by use of the 16th order system.

BACC representations were computed using the same DFT-based bandpass filters except that the triangular weighting functions for the MFSC were replaced by weighting functions with the frequency response of the Seneff bandpass filters. We considered two versions of the PLP as well. The default version was exactly as described by Hermansky in the literature[43]. The second realization of the PLP front end used a slightly different cepstral weighting factor which we found to be promising in previous pilot experiment on a subset of the AN4 training data.

Results obtained using these front ends are summarized in Table 5-2, with training using the CLSTK microphone. When the CLSTK microphone was used for testing, best recognition rates of

Processor Version	Language Weight	CLSTK Testing	CRPZM Testing
MFCC 12/40	5.5	84.4%	27.4
	7.5	82.8	30.3
MFCC 12/53	6.5	84.5	31.5
	9.0	82.7	33.9
MFCC 16/40	5.0	84.3	29.7
	9.0	82.0	35.0
BACC 12/40	4.0	82.7	15.5
	9.0	79.9	28.6
PLP (S=0.6)	7.0	82.0	26.0
PLP (S=0.7)	6.0	83.5	26.2
	8.0	81.3	28.5

Table 5-2 Comparison of results obtained using various physiologically-motivated front ends for the AN4 task. Three different implementations of MFCC were considered, varying the mel-scale channel spacing of the BPFs (MFCC 12/40 and MFCC 12/53) and the number of cepstral coefficients (MFCC16/40 and MFCC12/40). In testing using the PLP representation, the default set of analysis parameters was used, except that a peak enhancement factor of S=0.7 used.

around 82~84% were obtained by several modules, which is close to the recognition accuracy of about 85% obtained using the standard front end. Using the CRPZM microphone for testing, none of the front ends considered performed as well as Seneff's GSD outputs as described previously.

It appears that the physiologically-motivated ideas incorporated in these DFT-based modules do not provide environmental robustness for raw feature parameters. The major difference between the GSD output of the Seneff model and the DFT-based front ends is in the nature of the information preserved. With the GSD, the relevant information is the periodicity of the signal and it is detected in the waveform domain, whereas in DFT-based processing, the relevant information is the short-term energy in a narrow range of frequencies. In principle, the output from the GSD front-end is not susceptible of spectral tilt owing to the detector's self-normalizing nature. On the other hand, as the raw feature, the energy spectra of DFT-based processing are subject to the spectral tilt; so are the derived cepstra.

5.6. Effect of Artificially-Added Noise and Spectral Tilt

We now consider the effects of artificially-added noise on recognition accuracy. Table 5-3 and Table 5-4 show results obtained for the AN4 speech recognition task under simulated noise conditions, using the LPCC front end with and without CDCN, and using Seneff's auditory front-end. Again, when the CLSTK microphone is used for both training and testing, the conventional LPCC-derived signal processing technique works the best, with the mean-rate outputs of the Seneff front end providing somewhat better results than the GSD outputs. As the testing environment becomes noisier, the system using the LPC-based processing suffers a sharp degradation in recognition accuracy. This trend is noticeable even for the relatively benign condition of +30 dB global SNR. The use of CDCN improves noise robustness, at least for SNRs of +20 dB and greater. The performance degradation with decreasing SNR is very rapid for LPC processing, but more gradual using CDCN. The GSD algorithm is less effective than others in the quiet recording environment, but it outperforms the rest in very noisy environment. Below +20 dB global SNR and worse, the mean rate and GSD outputs of the Seneff's auditory front-end provide better recognition accuracy than the LPCC-based approaches.

Front-End	Noisy CLSTK Testing / CLSTK Training			
	Clean	30 dB	20 dB	10 dB
LPCC	85.7%	69.2	24.0	5.3
LPCC+CDCN	80.7	79.9	68.3	35.6
Mean-Rate	80.0	78.1	74.1	36.6
GSD	75.7	73.1	67.9	41.4

Table 5-3 Comparison of recognition accuracy with simulated additive noise, using LPCC with and without CDCN pre-processing, and using the mean-rate and GSD outputs of the Seneff model. The CLSTK microphone was used for both training and testing. A language weight of 7.0 was used.

Of the results obtained using the CRPZM microphone, LPCC processing with CDCN provided the best performance. Both the mean-rate and GSD parameters work better than the original LPCC-based features in most conditions, showing similarly consistent trend observed in the above cases, however, none of them outperformed the CDCN in this series of experiment. Also, the results show that the GSD provides better results than the mean-rate outputs for all conditions examined. These findings suggest that when training and testing environments are mismatched, individual use of either the mean-rate or GSD outputs may not provide as much correction for spectral tilt as the CDCN algorithm. Nevertheless, the gradual degradation in recognition accuracy with decreasing SNR over a wide range is observed consistently, independently of the type of testing microphone used

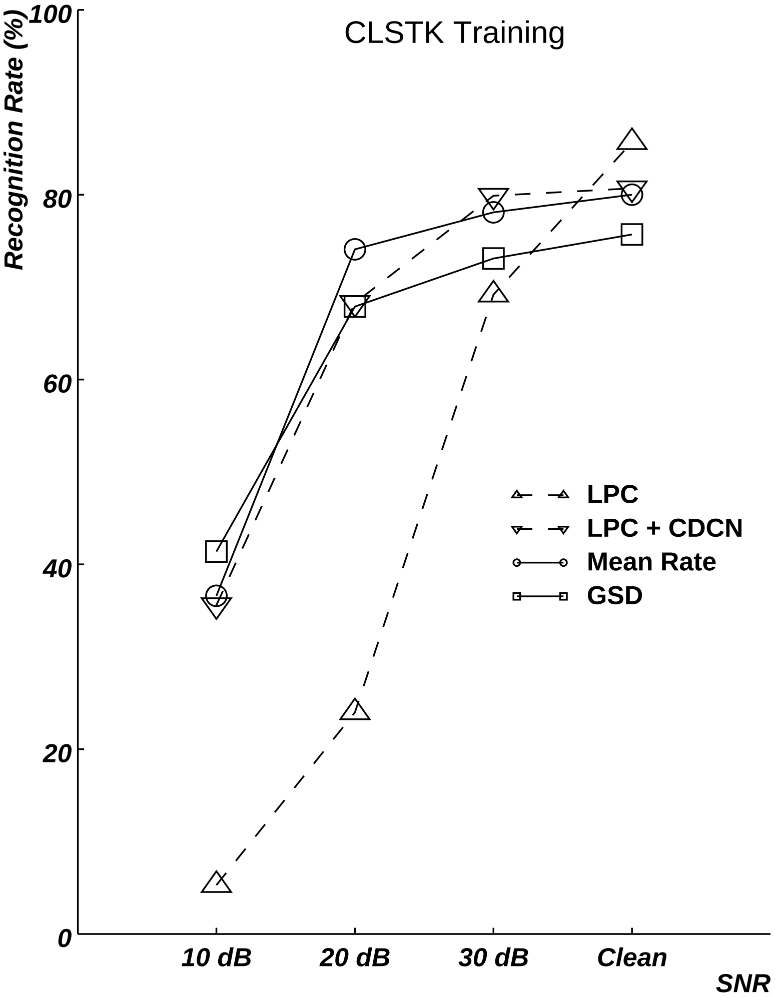

Figure 5-1 Comparison of recognition accuracy with simulated additive noise, using LPCC with and without CDCN pre-processing, and using the mean-rate and GSD outputs of the Seneff model. The CLSTK microphone was used for both training and testing. A language weight of 7.0 was used.

We also investigated the difference between recognition accuracy obtained using as the masker white noise versus speech-babble noise, as was used in Jankowski’s study [46]. We expected that babble should be a more difficult interference than white noise, as it is in fact speech jamming speech. However, surprisingly, white noise caused more severe degradation than babble did when testing using the CLSTK microphone, as shown in Figure 5-4.

In the last series of artificially-added degradations, we conducted an experiment using filtered speech.We passed the AN4 test set through a high-pass filter with system function $H(z) = 1 - 0.98z^{-1}$ and obtained the results summarized in Table 5-5. Compared with the LPCC which was very sensitive to the spectral tilt, both mean-rate and GSD parameters were found to be robust as raw feature parameters. GSD parameters were found to be relatively robust in the presence of spectral tilt, and CDCN compensates for linear filtering effect most successfully. There was a trend between the CRPZM data and the filtered CRPZM data. The better results obtained using

Front-End	Noisy CRPZM Testing / CLSTK Training			
	Clean	30 dB	20 dB	10 dB
LPCC	27.3%	34.0	21.2	3.3
LPCC+CDCN	71.0	66.4	49.2	21.9
Mean-Rate	32.3	31.2	23.8	6.4
GSD	48.7	48.1	29.4	12.2

Table 5-4 Comparison of recognition accuracy obtained using LPCC with and without CDCN pre-processing, and the mean-rate and GSD outputs of the Seneff model with simulated additive noise. The system was trained using the CLSTK microphone and tested using the CRPZM microphone. A language weight of 7.0 was used.

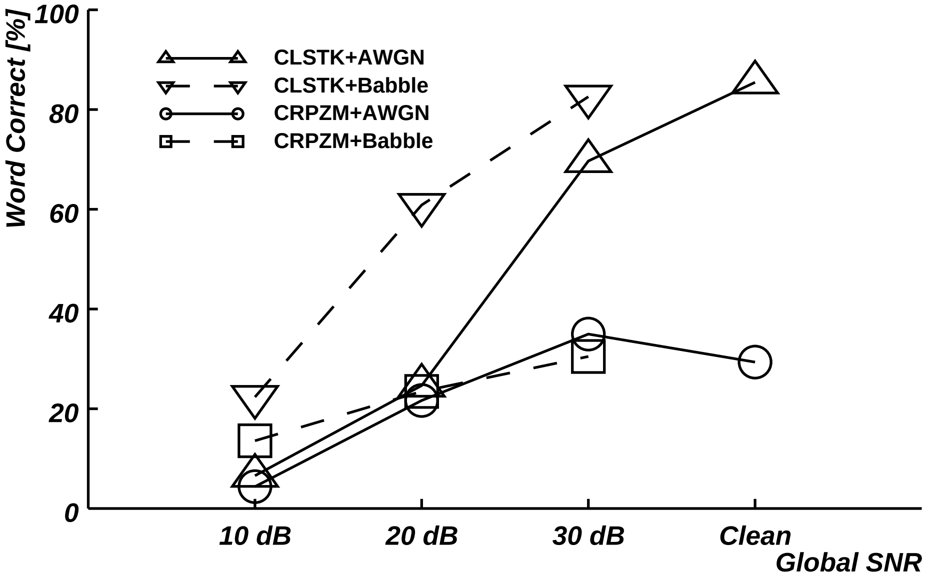

Figure 5-3 Effect of type of additive noise. White Gaussian noise (triangles and circles) and speech babble noise (dels and boxes) were added to simulate degraded environment at the global SNR of 10dB, 20dB and 30dB. The system was trained using the CLSTK microphone and tested using the microphone indicated. The LPCC front-end was used for this study.

filtered CRPZM data could be attributed to the low-pass nature of the CRPZM data compared with the CLSTK data.

A summary of this section is as follows:

1. LPCC with the CDCN was most successful in handling both additive noise and linear filtering.
2. As raw feature parameters, both mean-rate and GSD are more robust than LPCC, and the performance degradation was more gradual than LPCC.

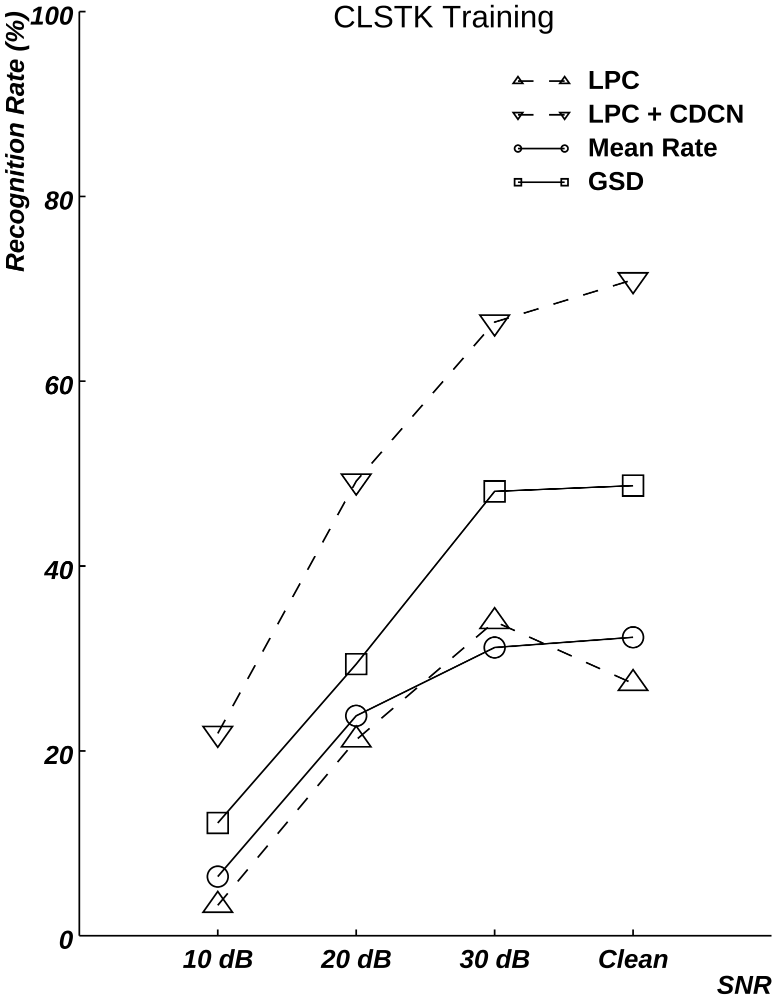

Figure 5-2 Comparison of recognition accuracy obtained using LPCC with and without CDCN pre-processing, and the mean-rate and GSD outputs of the Seneff model with simulated additive noise. The system was trained using the CLSTK microphone and tested using the CRPZM microphone. A language weight of 7.0 was used.

Front-End	Testing Data			
	CLSTK+HPF	CLSTK	CRPZM+HPF	CRPZM
LPCC	32.9%	85.3	27.2	18.6
LPCC+CDCN	78.6	82.2	64.0	71.5
Mean-Rate	61.5	80.0	45.6	32.3
GSD	71.9	75.7	49.0	48.7

Table 5-5 Comparison of acoustical pre-processing and auditory processing under simulated spectral tilt conditions. The system was trained using the CLSTK microphone and tested as indicated. A language weight of 4.0 was used for testing with LPCC parameters, and 7.0 was used for testing with the mean-rate and GSD outputs.

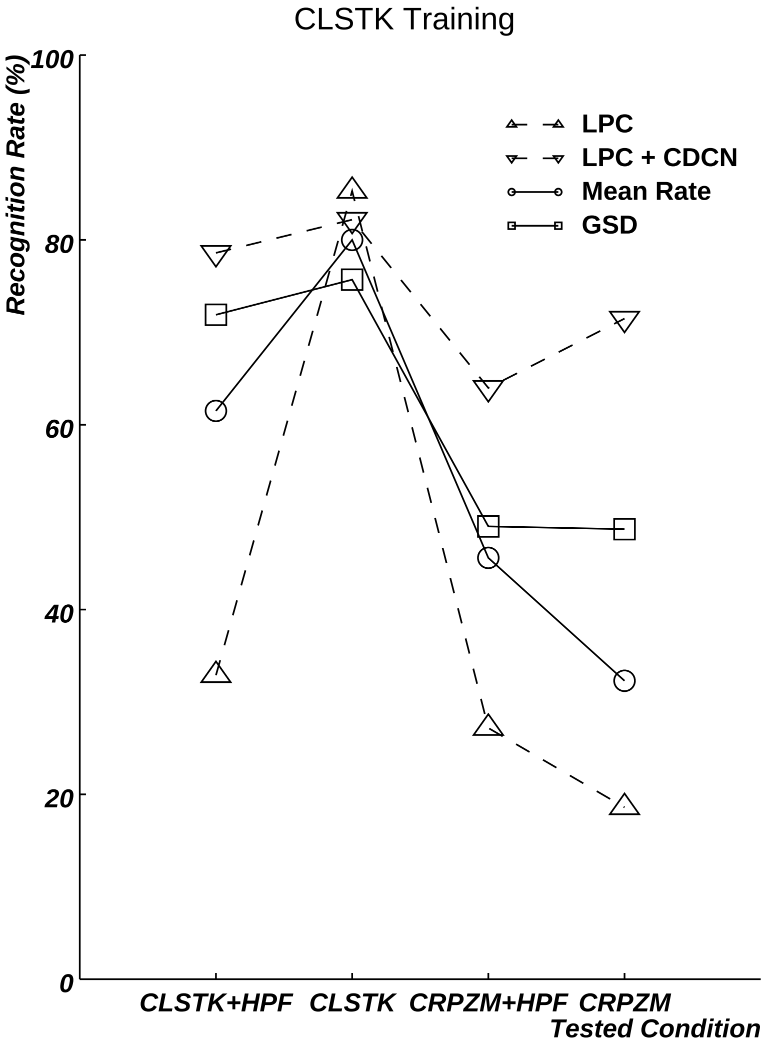

Figure 5-3 Comparison of acoustical pre-processing and auditory processing under simulated spectral tilt conditions. The system was trained using the CLSTK microphone and tested as indicated. A language weight of 4.0 was used for testing with LPCC parameters, and 7.0 was used for testing with the mean-rate and GSD outputs.

3. White noise caused more degradation than speech babble.

5.7. Effect of Language Weight

Any modification made to the front end results in feature vectors of a different statistical nature. This means that not only must the whole system be retrained using the new feature parameters, but also that the optimal operating conditions must be re-evaluated for the decoder.

In the 3-codebook discrete HMM version of the SPHINX recognition system, there are two system parameters that can be adjusted to optimize the decoder's behavior, the insertion penalty and the language weight. Of these two free parameters in the feature-extraction process, the language weight has been empirically found to be the more important, which is designed to control the relative impact of constraints imposed by the statistical language model.The decoder is optimized by searching empirically for the language weight that provides the greatest recognition accuracy. In general, one would like to have the importance of the language model to be greater when the acoustic models are not very reliable. Conversely, when phonetic hypotheses can be developed from the speech input with high confidence, relatively little attention need be paid to constraints imposed by the language model. Therefore, the best choice of language weight depends at least in part on the SNR to be expected.

To examine the effect of language rate on recognition accuracy, we compared the recognition accuracy obtained using front ends based on conventional LPCC-based processing, and using the mean-rate output of the Seneff model. In each case, data were obtained using the same task, but a wide range of language weights. Figure 5-4 and Figure 5-5 describe the dependence of recognition accuracy on language weight using the LPCC front-end and the AN4 task.

As can be seen in Figure 5-4, raw recognition accuracy monotonically decreases with increasing language weight when the CLSTK microphone is used for testing. (*Raw recognition rate* is the score obtained without the additional “insertion penalty” that reflects the percentage of insertion errors) However, the commonly-accepted figure of merit within the ARPA community is recognition rate after penalizing for insertion errors; this statistic exhibits a non-monotonic curve. In the case of the LPCC front-end using the CLSTK microphone, the optimal language weight was found to be around 6.0. It was also empirically found that the number of insertion and deletion errors are comparable under such optimal operating conditions.

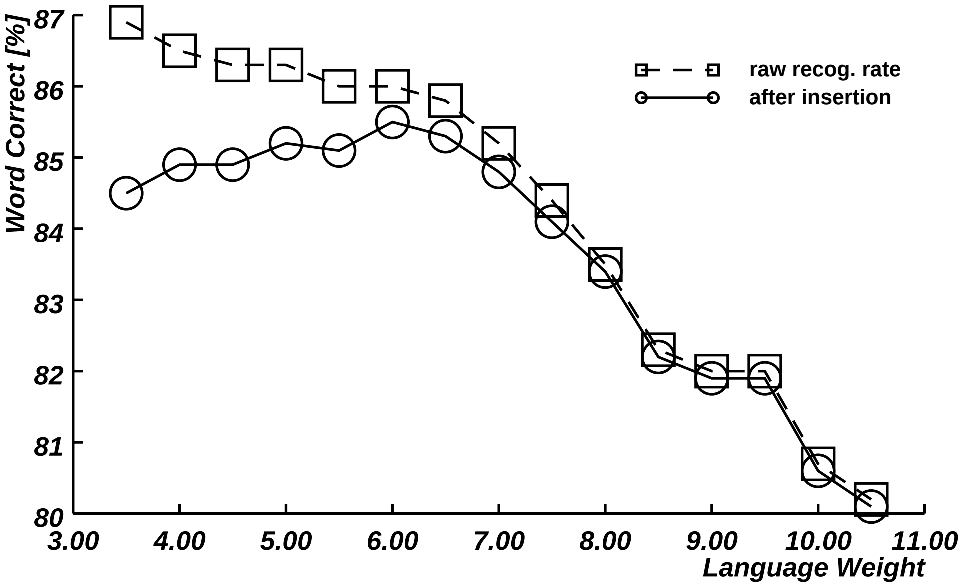

Figure 5-4 Language-weight dependency for the AN4 word recognition task using the LPCC front end. Plotted are raw recognition rates (boxes) and recognition rates after including the insertion penalty (circles), training and testing using the CLSTK microphone. The optimal language weight with insertion penalty was found to be 6.0.

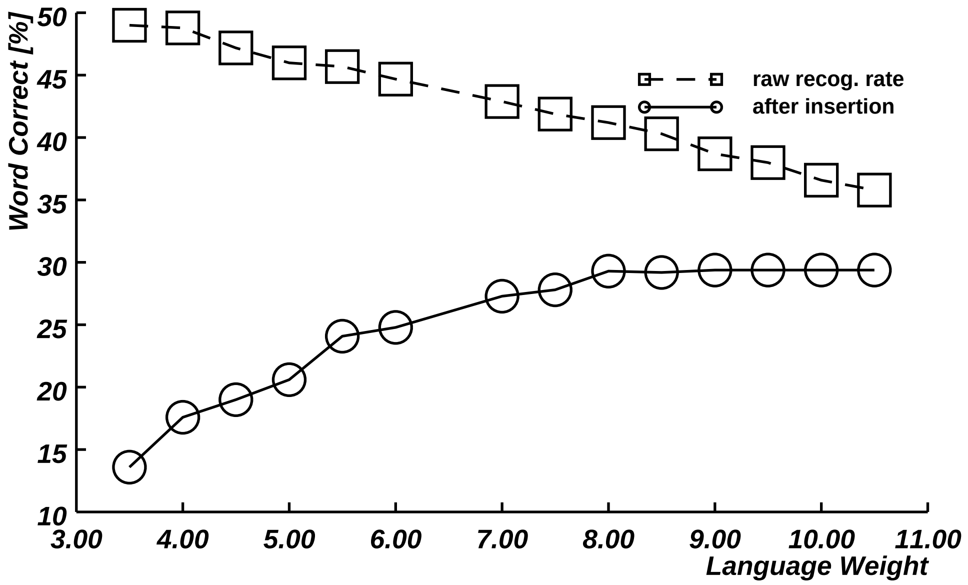

Figure 5-5 Language-weight dependency for the AN4 word recognition task using the LPCC front end. Plotted are raw recognition rates (boxes) and recognition rates after including the insertion penalty (circles), training using the CLSTK microphone and testing using the CRPZM microphone. The optimal language weight with insertion penalty was found to be around 9.0~10.5.

When the system is trained using the CLSTK microphone but tested using the CRPZM (Figure 5-5), we observe similar trends in both raw recognition rate and the recognition rate including the insertion penalty. The optimal language weight for this case is greater than the optimal weight for the case of training and testing using the CLSTK microphone. This is intuitively reasonable, as a mismatch between training and testing conditions tends to result in less reliable acoustic scores, which should increase the best language weight.

To confirm these trends, we also compiled similar results for the mean-rate and GSD outputs of the Seneff model, MFCC, BACC, and PLP. In each case, the general dependence of recognition accuracy as a function of language weight follows a similar trend. For example, results obtained using the mean-rate outputs of the Seneff front end are shown in Figure 5-6 and Figure 5-7, and they exhibit the same characteristics as the LPCC results. There were sometimes minor performance fluctuations, as can be seen in Figure 5-6 around the language weight of 6.0 to 6.5. However, most of these fluctuations are too small to be statistically significant. Hence, it is reasonable to conclude that there is a region of best language weights, if not a single point, that can be searched for each type of front-end processing.

To summarize, we observed a local maximum in recognition accuracy as a function of the language weight. While the best language weight can differ from one front end to another, and also from one testing environment to another, the characteristics of the performance curve are all similar. In general, the greater language weights produce better recognition accuracy when training and testing conditions are mismatched.

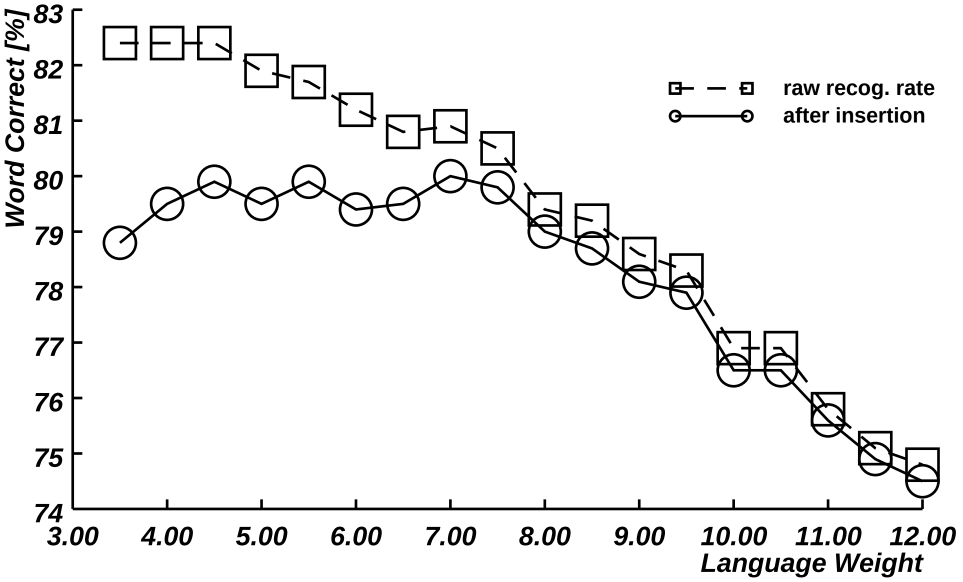

Figure 5-6 Language-weight dependency for the AN4 word recognition task using the mean-rate outputs of the Seneff model. Plotted are raw recognition rates (boxes) and recognition rates after including the insertion penalty (circles), training and testing using the CLSTK microphone. The optimal language weight with insertion penalty was found to be 7.0.

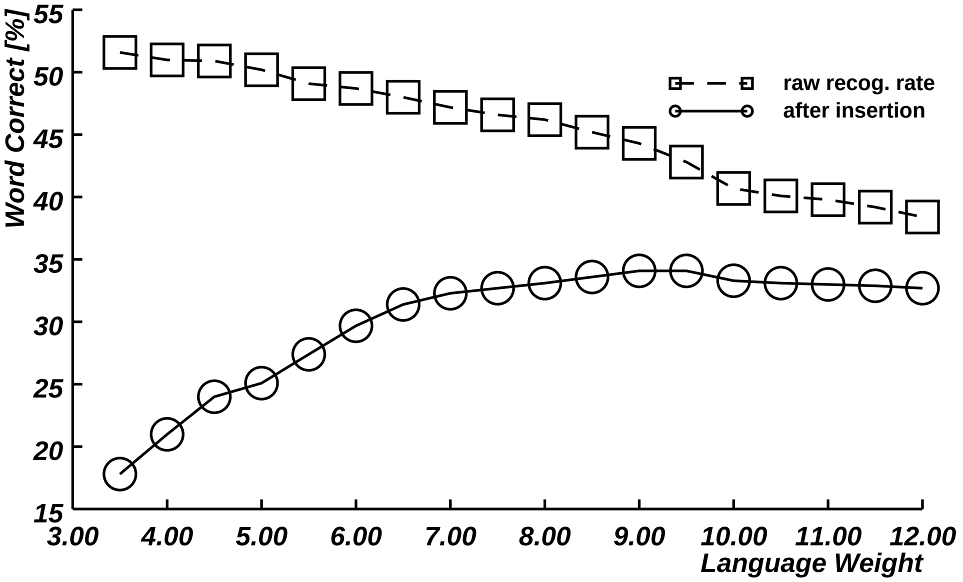

Figure 5-7 Language-weight dependency for the AN4 word recognition task using the mean-rate outputs of the Seneff model. Plotted are raw recognition rates (boxes) and recognition rates after including the insertion penalty (circles), training using the CLSTLK microphone and testing using the CRPZM microphone. The optimal language weight including the insertion penalty was found to be 9.0~9.5.

5.8. Feature Extraction of the Mean-rate and GSD Parameters

Some of the most important decisions to be made in the design of any pattern classification system include the selection of the set of features to be used by the classifier. Although one might intuitively expect that increasing the number of features will provide more useful information for the classifier, classification accuracy can deteriorate as more features are added if the amount of training data is limited. It is frequently the case that many more potential features can be provided by the initial signal processing than are needed for best classification performance.

Principal component analysis[47] is a very useful statistical procedure that is often useful in dealing with some unknown nature of a potential feature space. A linear transformation of the feature vector is defined such that the transformation diagonalizes the covariance matrix. As a result, the transformed vector consists of so-called principal components of the original feature vector. The basis used to diagonalize the covariance matrix forms the Karhunen-Loève (KL) transform. The dimensionality reduction can be achieved by selecting a small number of most significant features in the rotated feature space, and discarding the others. Until now there have been only a limited amount of attempts to use statistical analysis procedures such as principal component analysis to reduce the number of potential features obtained from an auditory-based front end. In this section, we compare recognition performance using the principal components of the Seneff-model output and we describe the degradation in recognition accuracy incurred when the number of features is decreased.

There are several considerations motivating this work. First, we would like to reduce the number of features to be used for classification from the 80 mean-rate and GSD outputs of the original Seneff model. (In contrast, the typical LPC-based speech system uses approximately 12~14 features.) We also believe that a procedure like principal components is likely to provide a set of features that is more robust than the original outputs of the Seneff model. Finally, we would like to be able to combine the mean-rate and GSD outputs of the Seneff model in a useful fashion. The “spectrum” produced by the ensemble of mean-rate outputs is broad, as it is a smoothed representation of the relative auditory-nerve firing rates. The GSD outputs, on the other hand, exhibit much more sharply-defined peaks.

The following procedure was taken to transform the AN4 training and testing data:

1. Compute the mean vector for the CLSTK training data, the CLSTK testing data, and the CRPZM testing data, and make them zero mean.

2. Load the covariance matrix Σ of the CLSTK training data, and find the transformation matrix A to diagonalize it by using its eigenvectors.

3. Transform all training and testing vectors by use of A

4. Use the first p principal components to form feature vectors of a reduced dimension.

We considered a number of different dimensions from the original 40 down to 2 for the principal components of mean-rate and GSD outputs, which produced the recognition accuracies summarized in Figure 5-8 and Figure 5-9, respectively: In addition to the standard experimental configuration using static features, their time-derivatives and the short-term energy information, we

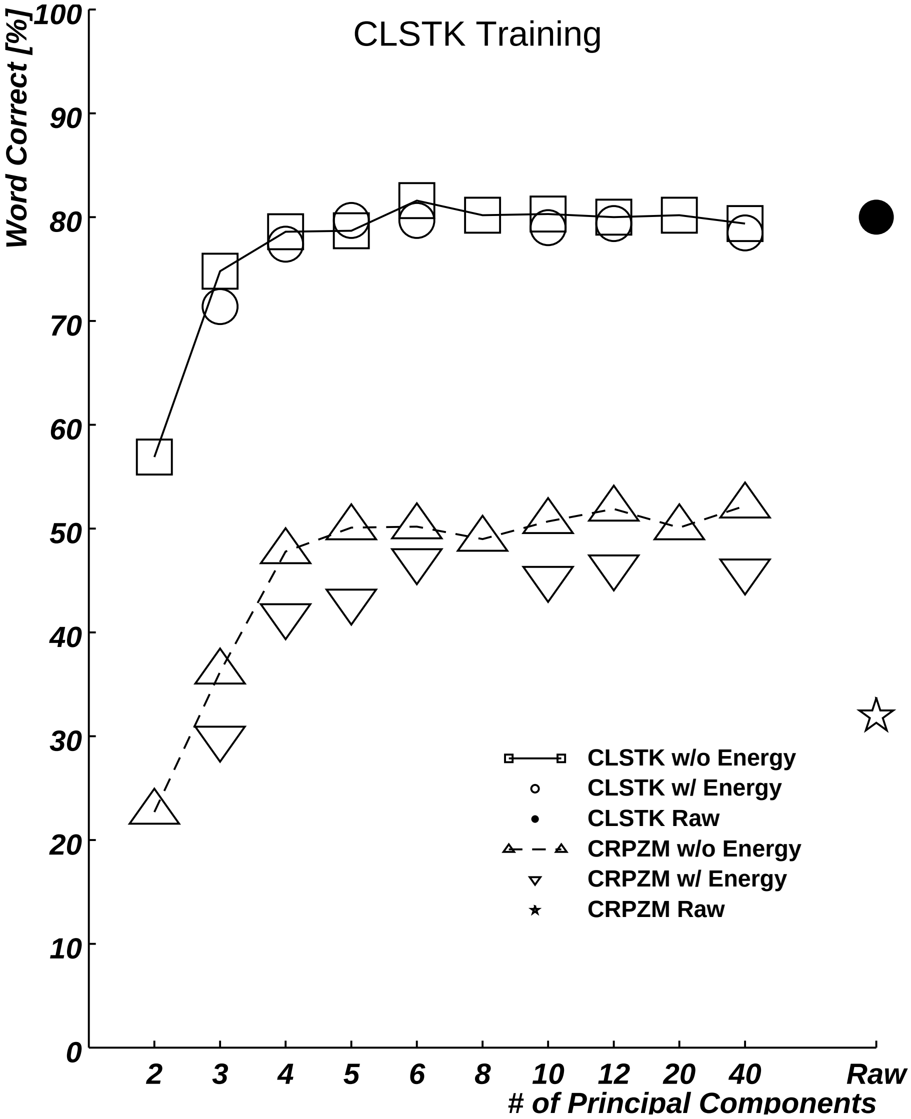

Figure 5-8 Dependence of recognition accuracy on the number of principal components derived from the mean-rate outputs of the Seneff model. The upper curves were obtained by testing using the CLSTK microphone, and the lower curves were obtained by testing using the CLSTK microphone. Better results were obtained by not using energy and difference energy features (boxes and triangles) than by including them with the default configuration (circles and dels). Circles and dels represent the use of the codeword corresponding to energy information. Also plotted are the baseline recognition rates using the raw features of mean-rate outputs (bullet and star).

conducted another set of experiments without using energy information*. Representative results from these comparisons are included in the same figures. In comparing systems, we concluded two tested systems to be similar if there was no difference in the composite report of all the significant tests by NIST benchmark programs.

*Energy information utilized by the 3-codebook SPHINX system is the composite distance measure of the frame energy and its time derivative. It should be noted that both static and dynamic information are implied whenever the energy information is referred to in the text.

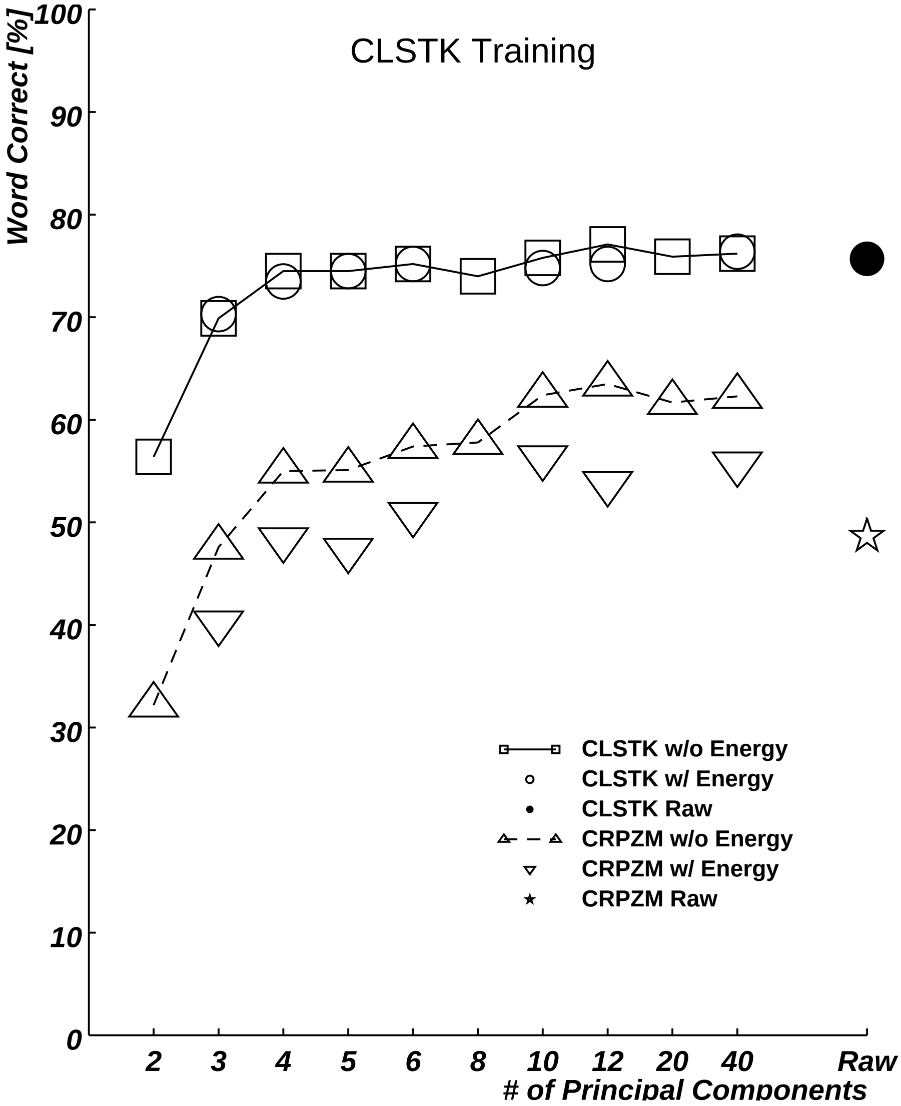

Figure 5-9 Dependence of recognition accuracy on the number of principal components derived from the GSD outputs of the Seneff model. The upper curves were obtained by testing using the CLSTK microphone, and the lower curves were obtained by testing using the CLSTK microphone. Better results were obtained by not using energy and difference energy features (boxes and triangles) than by including them with the default configuration (circles and dels). Circles and dels represent the use of the codeword corresponding to energy information. Also plotted are the baseline recognition rates using the raw features of mean-rate outputs (a bullet and a star).

When the CLSTK microphone is used for both training and testing with the mean-rate principal component system, we found no statistically significant difference in recognition performance for any number of principal components from the full set of 40 down to 4 when energy was not used. However, the systems with 2 and 3 principal components were found to be inferior to the rest.

There was also a small fluctuation among the best performing group. For instance, the 6-component system was reported to be better than 4, 5, 40 component systems in a matched-pairs test. When energy was included, the 5 principal components were required to maintain the performance of the full feature set.

In the CRPZM microphone testing, performance degradation started when the 4-component system was used, and smaller systems than the 4 were clearly inferior to the system with5 or more principal components. With the inclusion of energy, performance dropped significantly. We also found that the mean-rate principal components had advantage over their raw feature counterpart in that it improved the cross microphone performance from 32.3% to 52.2%, while there was no performance degradation in the same microphone testing.

In the experiments using GSD principal components, we observed that the dimension was successfully reduced from 40 down to 4 with no performance degradation in the CLSTK testing. In the CRPZM testing, however, the 10-component system was the most compact. Performance dropped significantly with 8 principal components or fewer. Similarly to the mean-rate case, significantly better results were obtained without the energy than by using the energy. The best performance of GSD principal components in the cross microphone testing was 63.5% using the 12-component system, as compared to the GSD baseline performance of 48.7%.

A summary of experimental results on the use of principal components of Seneff model outputs is shown in Table 5-6 and our finding is summarized as follows:

1. The use of the principal components improved significantly the cross microphone testing performance both with the mean-rate and GSD parameters, while there was no loss in performance for the same microphone testing.
2. The smallest size of the principal component vector was 5 for the mean-rate parameters and was 10 for the GSD parameters to be effective, which was compared to the original vector size of 40.
3. There was also a significant performance improvement obtained by discarding the energy information in the cross-microphone testing both with the mean-rate and GSD parameters, while there was no effect in the same microphone testing.

Front-End Version	Feature Vector Size	Maximum Performance Plateau	
		CLSTK	CRPZM
Raw Mean-rate	40	80.0%	32.3
Mean-rate Principal Components	6	78.5~79.7	45.2~46.9
Mean-rate Principal Components w/o Energy	5	78.7~81.6	49.0~52.2
Raw GSD	40	75.7	48.7
GSD Principal Components	10	74.8~76.4	53.8~55.7
GSD Principal Components w/o Energy	10	75.8~77.1	60.2~63.5

Table 5-6 Comparison of recognition accuracy obtained for the AN4 task using the outputs of the Seneff model outputs and their principal components. The performance plateau achieved by the best implementation for each configuration is described, using systems with a smaller number of parameters if possible is also represented.

5.8.1 Discussion

Jankowski[46] had previously investigated the effect of reducing the dimensionality of principal components of the Seneff model by using an HMM-based, isolated-word recognition system and concluded that the use of more coefficients would result in better recognition performance. The task was to recognize 105 words including digits and command words. In his experiment, both the mean-rate and GSD parameters maintained error rates of about 0.9~1.1%, while the parameter dimensionality was reduced from 40 down to 25. For feature vector dimensions of smaller than 20, he observed a gradual degradation in recognition accuracy. Our finding with the AN4 task, on the contrary, was that a relatively small number of principal components of the order of 6 would perform at a level comparable to the full feature set using either microphone

It should be noted that in our experiment the speaker groups for the training and the testing were disjoint ensembles of 74 and 10 speakers respectively, and therefore, the recognition system was evaluated against unknown speakers. In Jankowski's experiment, the training and testing tokens were disjoint data sets that were uttered by the same group of 8 speakers.

Jankowski achieved a high level of recognition accuracy using the full set of principal components, and the performance degradation he observed as dimensionality was reduced indicated that feature sets consisting of small number of principal components were not capable of representing the nature of the corpus he used. This could have been the reflection of that the system was well tuned to a small number of training/testing speakers and therefore detailed spectral information contributing more likely to the higher order principal components was also well utilized.

While the size and the contents of vocabulary in the AN4 task were somewhat similar to the task used by Jankowski, the highest recognition rate was about 85% using LPCC and 80~81% using the raw feature parameters of the Seneff model or their principal components. The use of a disjoint set of training and testing speakers forces the front end deal to contend with not only phonetically relevant information for the decoder but also other phonetically irrelevant information due to the spectral variability of unknown testing speakers.

Therefore, it is possible that the principal component analysis conducted for our experiment was particularly useful in achieving the desirable abstraction of spectral information needed to overcome spectral variability among speakers. In many applications of spoken language systems, it is necessary to not assume that the voice characteristics of users are known to the recognizer. Our findings suggest that the use of small number of principal components could successfully preserve the relevant information of input speech.

Lee[1] found the use of frame energy information to be helpful in the early development of the 3-codebook SPHINX system using the LPCC front end. His pilot experiments also revealed that the time derivative of frame energy provided greater improvement than frame energy. The temporal profile of the short-term energy and its derivative made it possible to achieve the more accurate segmentation of words in conjunction with cepstral parameters of which magnitude is independent of the signal energy.

Unlike cepstral parameters, the mean-rate parameters do correlate with the input energy and reflect it in a somewhat compressed manner. The GSD parameters also seem to have some correlation with the short-term energy of the input signal. The relationship for the GSD outputs is less obvious than that of the mean-rate outputs, and somewhat indirect as the GSD outputs were designed to detect channel synchrony of the signal in a normalized manner rather than to be a compressed energy measure.

A similar property should also be found in the principal components of the mean-rate and GSD outputs, as they are results of a linear transformation of the original features. Hence, the magnitude relationship between corresponding frames are preserved. When testing with the CLSTK microphone, the fact that the presence or absence of explicit energy information did not affect performance may imply that no new information about the speech signal is provided by the energy term. We do not have a very clear explanation to the cross-microphone result, although it was suspected that the energy information computed in the CRPZM testing, which was a mismatched environment, might have misguided the decoder, which would had performed better otherwise.

5.9. Combination of Principal Components of the Mean-rate and GSD Outputs

As we have described in the previous chapter, the mean-rate output provides a smoothed short-term representation of the firing probability of fibers of the auditory nerve and the GSD outputs describe the periodicity with respect to a given channel's CF. They are of a complementary nature to some extent, and the combination of the two might extend the capability of front-end processing. Meng and Zue [6] recently demonstrated that the combined feature set formed by principal component vectors of the mean-rate and GSD parameters outperformed both the original Seneff model parameters and conventional front-end outputs such as the MFSC, MFCC, and DFT in a vowel recognition task of clean and noise injected speech.

We considered two different ways of combining the mean-rate and GSD outputs. Our first approach was to duplicate Meng and Zue's procedure, concatenating the first 20 principal components of both the mean-rate and GSD outputs to provide a new vector of size 40. As an alternative approach, we integrated both principal component parameters at the codeword level rather than merging them at the feature vector level.Table 5-7 shows how we used three codewords to represent the mean-rate and GSD features.

	Original	Alternative 1	Alternative 2
Codeword1	Static Feature	Static Mean-rate Principal Components	Static GSD Principal Components
Codeword2	Dynamic Feature	Dynamic Mean-rate Principal Components	Dynamic GSD Principal Components
Codeword3	Composite Energy (Static + Dynamic)	Static GSD Principal Components	Static Mean-rate Principal Components

Table 5-7 The codeword sets used in combining mean-rate and GSD information.

Instead of using three VQ codebooks representing the static and dynamic features of principal components and the composite energy information, we used them for the static features of mean-rate and GSD principal components and the dynamic features of either mean-rate or GSD principal components. All VQ codewords were generated by clustering principal-component vectors of size 40, except that the codebook for the dynamic features of mean-rate principal components was computed using vectors of size 6. We compared these results to the performance of systems that combined only static features of 6, 12, and 40 mean-rate and GSD principal components.

Figure 5-10 compares the recognition accuracy obtained using the combination of principal components of the mean-rate and GSD outputs against the best performance achieved by use of principal components from the mean-rate and GSD outputs considered in isolation. We found that the combined principal components performed well when testing with the CLSTK microphone, but that this type of feature set was not successful when testing with the CRPZM microphone. On the other hand, the codeword-level integration showed good performance in both testing conditions.

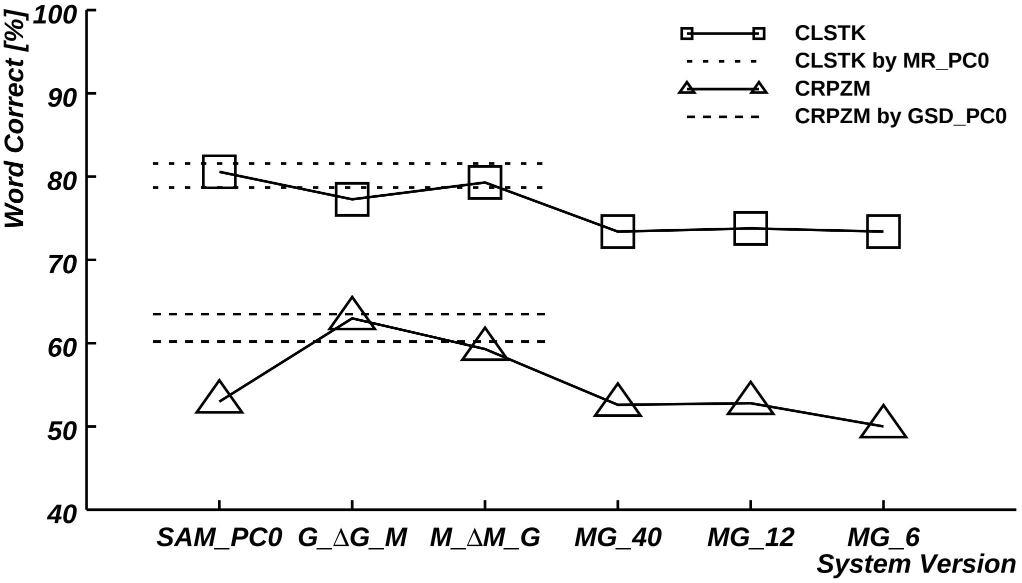

Figure 5-10 Comparisons of results of attempts to combine principal components of the Seneff model in the AN4 word task. System configurations include parameter-level integration of the principal components of the Seneff auditory model (SAM_PC0), and codeword-level integration of static and dynamic features of the mean-rate and GSD principal components (G_ΔG_M and M_ΔM_G). The complete principal-component vectors of size 40 were used for all results except that ΔM was taken from a mean-rate principal-component vector of size 6. Also plotted are results obtained using static features only. The suffix denotes the size of the feature vectors used (MG_40, MG_12, and MG_6). Dashed lines in the plot show the range of performance achieved by the mean-rate principal components when testing with the CLSTK microphone, and using GSD principal components when testing with the CRPZM microphone.

By including the static features of the mean-rate principal components, the system using the principal components of the GSD outputs achieved a recognition accuracy of 63.0% when testing with the CRPZM microphone, which was the level of performance achieved by using GSD principal components. When testing with the CLSTK microphone, the combination of principal components also performed reasonably well compared with the best results by mean-rate principal components. The mean-rate system with static GSD principal component features showed the opposite trend and was more successful in the CLSTK testing (79.5%) than in the CRPZM testing, measured relative to the best performance obtained in the isolated use of either mean-rate or GSD principal components. Note that the recognition rate was 49.0~52.2% by mean-rate principal components in the CRPZM testing, which was brought up to 59.3% by utilizing static GSD principal components.

Systems without the use of dynamic features performed poorly. There was 3.5~5.9% degradation in the CLSTK testing and 6.5~13.0% in the CRPZM testing, compared with the ones using the dynamic features.

In summary, we found that the principal components of the mean-rate and GSD outputs can be successfully combined at the codeword level. We believe that this combination can compensate for some of the weaknesses of principal components of the mean-rate outputs, while preserving their strength. Nevertheless, the combination of the two sets of principal components did not improve the performance of the mean-rate baseline system when testing using the CLSTK microphone.

There appears to be no particular merit in combining the 20 principal components of the mean-rate and GSD outputs in the parameter domain.

Because the use of the combined sets of components does not exceed baseline performance derived from mean-rate output alone, we did not pursue this issue at the feature vector level. While there may be some better combination of principal components of mean-rate and GSD outputs, differences between Meng and Zue's study and our findings must be attributed to differences in the classifier. Being a discrete HMM system, SPHINX performs best when the decision regions are well characterized by the distance metric. In their evaluation, Meng and Zue used a multi-layer Perceptron, which is capable of training more difficult decision regions that Gaussian classifiers fail.

The simple concatenation of the two different types of feature vectors forms a complicated parameter space, with characteristics that are not well represented by conventional distance metrics. It is possible that sets of features that are useful when considered in isolation may become obscured when are combined if the distance metric used to compare stored templates to the incoming feature vectors is inappropriate. Similarly, dynamic features of these parameters, such changes in the mean-rate and GSD principal components over time might cancel out and become undetectable even though they may be useful if considered individually.

The integration of two features at the codeword level provides us with a more suitable utilization of desirable information. In the 3-codebook SPHINX system, information from three different sources is combined in the form of the equally weighted sum of three log-likelihood parameters computed based on the sample statistics of each codeword. As a result, differences in statistical nature among three codewords are preserved and treated as having equal contribution at the training of HMMs and also at the decoding in computing the acoustic scores.

5.10. Summary

In summary, the baseline recognition accuracy of the LPCC front end for the AN4 word recognition task was 85.3% using the CLSTK microphone, while the mean-rate and GSD produced accuracies of 80.0% and 75.7% respectively. The outputs of the GSD provided the best performance

(48.7% correct) when the PZM6FS microphone was used for testing, followed by the mean-rate output (32.3%) and LPCC processing (18.6%).

The front ends that used MFCC, BACC, and PLP features all performed similarly to LPCC when the CLSTK microphone was used. Increasing the BPF channel spacing or the number of cepstrum parameters had no significant impact. When testing using the CRPZM microphone, neither MFCC, BACC, nor PLP were as successful as the GSD was, achieving an intermediate level of performance between the GSD and the LPCC.

When noise was artificially added to the speech signal, both mean-rate and GSD outputs of the Seneff model provided an improvement in recognition accuracy compared to the original LPCC-based processing, especially when the SNR was low. Similar results were observed when the training and testing sets of speech samples exhibited different spectral tilt. The LPCC-based front end combined with CDCN, however, outperformed both sets of outputs of the Seneff model, especially when the SNR was low or different microphones were used for training and testing. We also found that white Gaussian noise caused a greater degradation in recognition accuracy than noise consisting of speech babble when presented at the same global SNR level.

A common dependence of recognition accuracy on language weight was observed for all front-end modules considered. The optimal language weight changes whenever the type of initial signal processing, or the training or testing environment was changed.

Finally, we found that the use of mean-rate and GSD principal components, improved the recognition accuracy of the Seneff model performance in difficult environments. The system achieved accuracies of 49.0~52.2% using the mean-rate outputs, and accuracies of 60.2~63.5% using the GSD outputs. The number of principal components needed to guarantee the above performance is 5 for the mean-rate outputs and 10 for GSD outputs. The explicit inclusion of energy information degraded the recognition accuracy of the Seneff model using principal components as the features. The mean-rate and GSD principal component features were successfully combined at the codeword level, but a similar combination in the feature-vector domain was not helpful. By using the static and dynamic feature of one front-end and adding the static feature of the other, weaknesses of the former are ameliorated.

Chapter 6 Significance of the Components of the Seneff Neural Transmitter Model

The functional components of the Seneff model were originally developed and refined to provide a pragmatic description of the response of peripheral auditory-nerve fibers to speech sounds, without directly determining the effect that these components might have on speech recognition accuracy. Similarly, it is not clear how well individual components of the model serve to preserve the representation of a speech sound in the presence of distortion introduced by additive noise or linear filtering. In this chapter we describe the results of a series of experiments that attempt to determine how the presence or absence of the individual components of the neural transmitter (NT) components of the Seneff front end affect the recognition accuracy of the SPHINX system. The underlying motivation of this work is to determine whether the computational complexity of the model (and others like it) could be reduced by eliminating some of its functional components without adversely affecting recognition accuracy.

While linear bandpass filters in physiologically-motivated models play an important role in processing speech sounds, their behavior has been extensively analyzed in the literature. Hence we will pay greatest attention to the components that provide the nonlinear transduction at the level of the hair cells. We have chosen the Seneff model as representative, as it contains the most essential elements of NT processing that are commonly found among many of previously proposed models, and because a detailed description of it was available at the time this work began.

We also consider the computational complexity incurred by individual components of the NT model. This is important as the total amount of computation of physiologically-motivated processing is currently much greater than the computational cost of conventional signal processing. We would like to become able to build a robust front end in a more cost-effective manner.

In the following sections, we first discuss the experimental procedure and results of experiments in which we assessed the importance of various components of the NT model by eliminating them from the system. We conclude with a discussion of computational complexity.

6.1 Functional Significance of Components of the NT Model

In Chapter 4, we discussed some of the effects of the NT processing on the spectral image of the incoming speech. In this section we discuss the results of experiments intended to evaluate the relative significance of the components of the NT mode using the effect on the recognition accuracy of the SPHINX system as the figure of merit for an individual model component.

The general procedure for this evaluation is to recover an individual component of the NT model from the sequence of front-end processing steps, and train and test the SPHINX system on AN4 data to observe the impact of eliminating that component. The system was trained using the AN4 close-talk microphone (CLSTK) with the various modified versions of Seneff front-end, and tested using both the same microphone and omnidirectional Crown PZM6FS microphone (CRPZM). Later, white Gaussian noise was added to testing data from both the CLSTK and CRPZM microphones at three different levels of 10, 20, and 30 dB global SNR.

Table 6-1~Table 6-3 and Figure 6-1~Table 6-3 summarize the results of the evaluation of the three major NT functions, short-term adaptation, the AGC, and the synchrony fall-off (see Figure 4-3). In each case we compare the recognition accuracy achieved using the conventional LPC-derived processing, results using the mean-rate and synchrony outputs of the Seneff processing with the complete NT model, and similar results with the short-term adaptation, AGC, and/or lowpass filtering disabled.

6.1.1. Short-term Adaptation and Rapid AGC

The result of the combined effects of short-term adaptation and the rapid AGC is the enhancement of the transient part of the input stimuli. Two different issues are associated with this processing: (1) reduction of the effect of background noise, and (2) enhancement of contrast between transient speech segments (such as onsets, bursts, and formant transitions) and sustained steady-state speech segments (such as vowels).

6.1.1.1. Short-term Adaptation

Results obtained by disengaging the short-term adaptation component of the NT model are designated by the symbol *-Adapt*. When the mean-rate parameters were used, we observed a significant drop in recognition performance by eliminating the adaptation stage, except for the two least noisy conditions testing using the CLSTK microphone. However, using the GSD outputs, no loss of accuracy was observed when adaptation was removed from the processing, with the single exception of the 10-dB SNR using the CRPZM microphone.

By comparing the results using the CLSTK and CRPZM testing microphones, we note a few characteristic trends in the performance regardless of whether the mean-rate or the GSD outputs are used:

1. For SNRs of +20 dB or greater, the omission of short-term adaptation does not cause significant degradation in recognition performance when training and testing using the CLSTK microphone.
2. Eliminating the adaptation stage results in an unacceptable performance degradation for all SNRs when testing using the CRPZM microphone.

To conclude, short-term adaptation is an important element in noisy conditions, but its value is not obvious for clean speech.

.

Front-End	Clean	30 dB	20 dB	10 dB
LPCC	85.7%	69.2	24.0	5.3
Mean-Rate(MR) Full NT	80.0	78.1	74.1	36.6
MR -Adapt.	80.0	78.6	62.2	10.6
MR -AGC	80.9	80.7	74.2	44.9
MR -LPF	79.5	79.0	72.2	37.1
MR -AGC-LPF	79.1	77.0	39.7	13.0

Table 6-1 Evaluation of the NT components for the AN4 task using the mean-rate outputs of the Seneff mode testing with the CLSTK microphone. Full NT denotes the original Seneff model in which all the NT functions ar enabled, while -Adapt, -AGC, -LPF, and -AGC-LPF denote that the corresponding NT functions are disengage respectively. The LPCC results are also listed for comparison. A language weight of 7.0 was used for all results.

Front-End	Clean	30 dB	20 dB	10 dB
LPCC	27.3%	34.0	21.2	3.3
Mean-Rate(MR) Full NT	32.3	31.2	23.8	6.4
MR -Adapt.	25.5	23.5	9.3	1.6
MR -AGC	32.7	32.0	26.2	10.9
MR -LPF	30.1	28.4	24.0	11.0
MR -AGC-LPF	34.0	28.7	17.7	6.6

Table 6-2 Evaluation of the NT components for the AN4 task using the mean-rate outputs of the Seneff model, testing with the CRPZM microphone. Full NT denotes the original Seneff model in which all the NT functions are enabled, while -Adapt, -AGC, -LPF, and -AGC-LPF denote that the corresponding NT functions are disengaged respectively. The LPCC results are also listed for comparison. A language weight of 7.0 was used for all results.

Front-End	Clean	30 dB	20 dB	10 dB
LPCC	85.7%	69.2	24.0	5.3
GSD Full NT	75.7	73.1	67.9	41.4
GSD -Adpt.	72.1	71.0	61.9	26.4
GSD -AGC	75.1	75.1	66.2	37.1
GSD -LPF	75.1	71.7	63.6	42.0
GSD -AGC-LPF	74.1	70.0	39.0	12.9

Table 6-3 Evaluation of the NT components for the AN4 task using the GSD outputs of the Seneff model, testing with the CLSTK microphone. Full NT denotes the original Seneff model in which all the NT functions are enabled while -Adapt, -AGC, -LPF, and -AGC-LPF denote that the corresponding NT functions are disengaged respectively The LPCC results are also listed for comparison. A language weight of 7.0 was used for all results.

Front-End	Clean	30 dB	20 dB	10 dB
LPCC	27.3%	34.0	21.2	3.3
GSD Full NT	48.7	48.1	29.4	12.2
GSD -Adpt.	25.9	26.5	24.1	15.2
GSD -AGC	50.0	51.4	31.5	10.8
GSD -LPF	40.0	38.7	23.1	8.1
GSD -AGC-LPF	43.5	37.3	19.2	7.2

Table 6-3 Evaluation of the NT components for the AN4 task using the GSD outputs of the Seneff model, testing with the CRPZM microphone. Full NT denotes the original Seneff model in which all the NT functions are enabled, while -Adapt, -AGC, -LPF, and -AGC-LPF denote that the corresponding NT functions are disengaged respectively. The LPCC results are also listed for comparison. A language weight of 7.0 was used for all results.

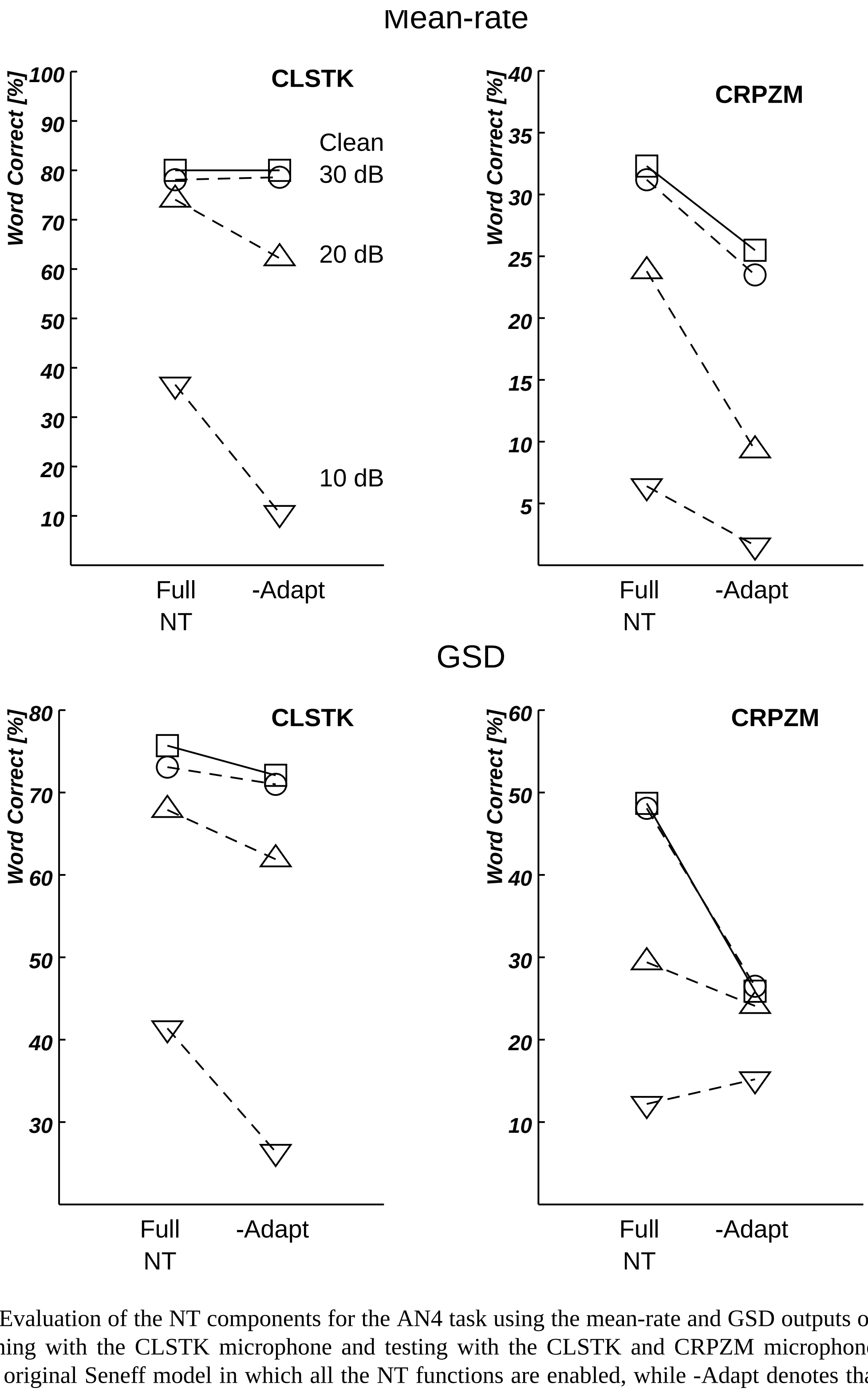

Figure 6-1 Evaluation of the NT components for the AN4 task using the mean-rate and GSD outputs of the Seneff model, training with the CLSTK microphone and testing with the CLSTK and CRPZM microphones. Full NT denotes the original Seneff model in which all the NT functions are enabled, while -Adapt denotes that the short-term adaptation NT function is disengaged.

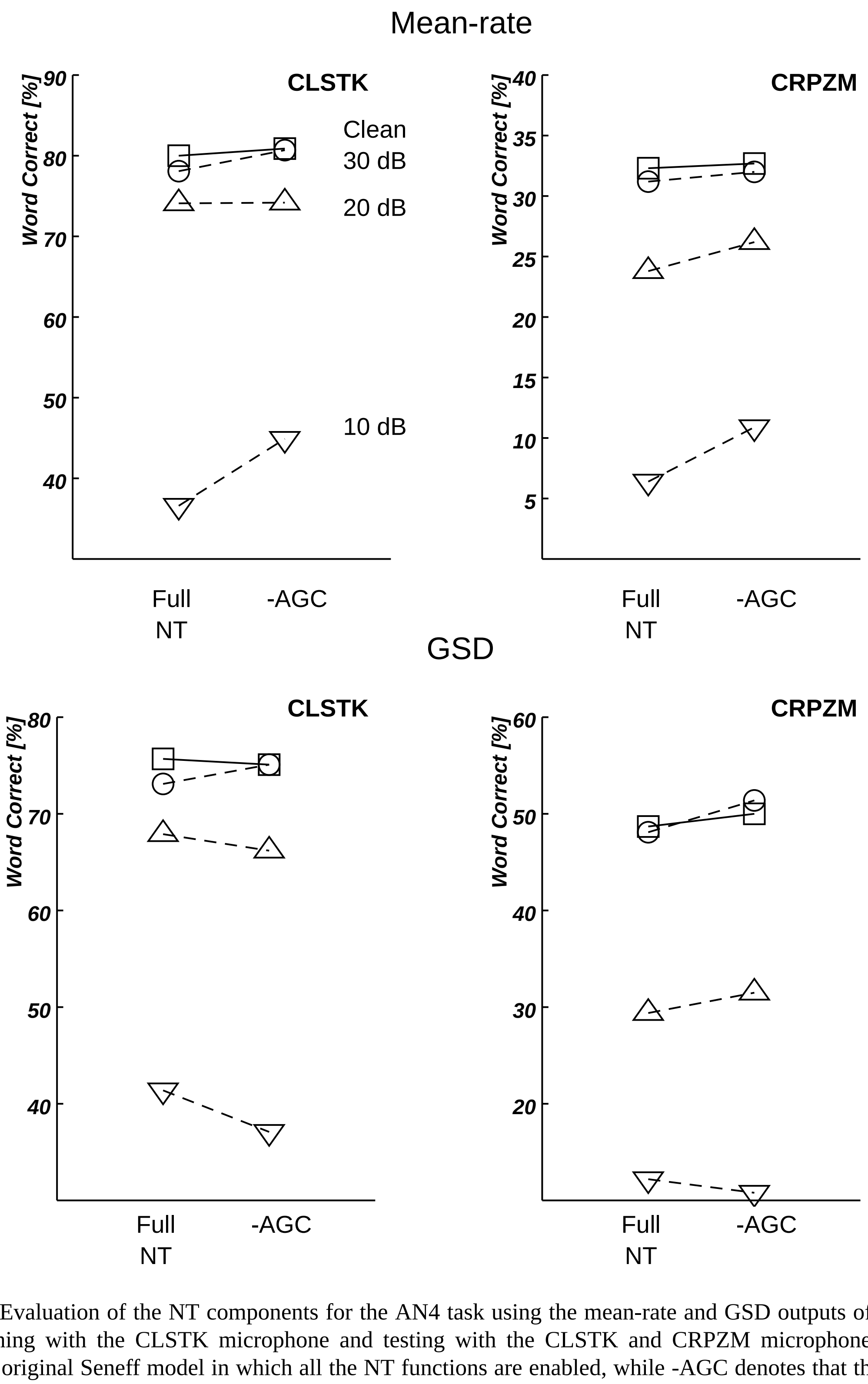

Figure 6-2 Evaluation of the NT components for the AN4 task using the mean-rate and GSD outputs of the Seneff model, training with the CLSTK microphone and testing with the CLSTK and CRPZM microphones. Full NT denotes the original Seneff model in which all the NT functions are enabled, while -AGC denotes that the AGC NT function is disengaged.

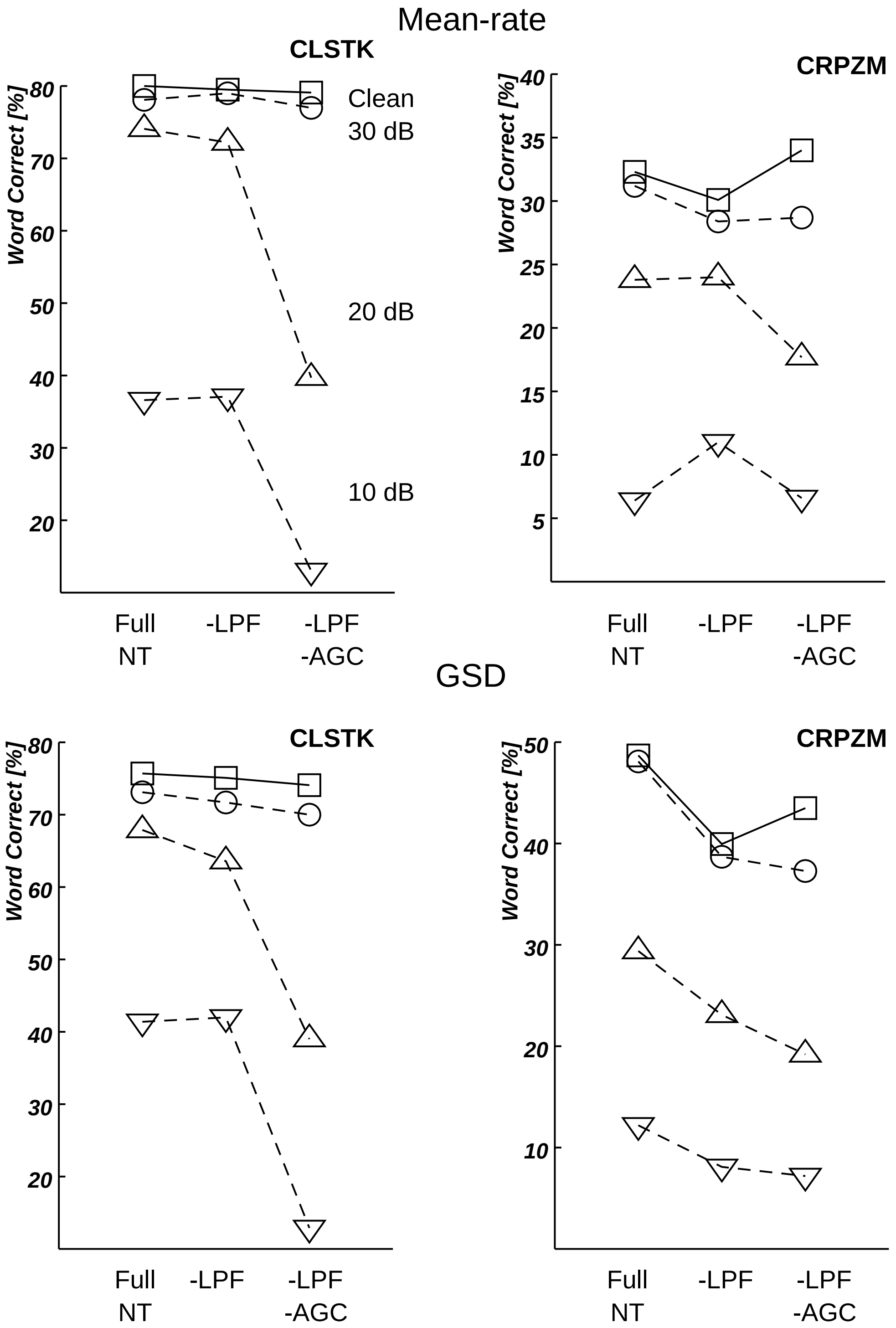

Figure 6-3 Evaluation of the NT components for the AN4 task using the mean-rate and GSD outputs of the Seneff model, training with the CLSTK microphone and testing with the CLSTK and CRPZM microphones. Full NT denotes the original Seneff model in which all the NT functions are enabled, while -LPF denotes that the LPF is disengaged. Likewise -LPF-AGC denotes both LPF and AGC are disengaged.

6.1.1.2. Automatic Gain Control (AGC)

For the most part, we found (to some surprise) that removing the AGC did not adversely affect recognition accuracy when the mean-rate outputs were used. This trend was also true for the GSD outputs, except for the lowest SNR testing with the CRPZM.

In general, the AGC sharpens the transient signal envelope of the outputs of each channel when the signal amplitude is very different from the running average of previous input signals, and suppresses the local fluctuations of the signal envelope, otherwise. This characteristic behavior is particularly useful in the human auditory periphery, since the human auditory system is only capable of handling a relatively narrow dynamic range of signals. However, for computational simulations in which the only limitation is word length, dynamic range reduction is not necessarily desirable, although the enhancement of transient envelopes may still remain beneficial when the SNR is very low. Therefore, it may be the case that the AGC component of the NT simulation is of limited value, and it must be carefully calibrated so as not to suppress the dynamic range too much. Except, perhaps for cases where the SNR is extremely low, the AGC could make a good candidate for omission, if the computational complexity has to be reduced.

6.1.2. Synchrony Fall-off

We have noted that the loss of synchrony at higher CFs is an important component in models of binaural hearing, but that such synchrony loss may not be particularly helpful for speech recognition. Hence we compared the recognition accuracy obtained using front ends with and without the LPF component included.

The experimental results indicate eliminating the LPF produces no degradation in recognition accuracy when the CLSTK microphone is used for testing, for both mean rate and GSD outputs. On the other hand, when testing using the CRPZM microphone, there was a 2% drop in recognition accuracy when the mean-rate outputs are used, and a greater difference when the GSD outputs are used.

We did not completely understand the reasons underlying the minor difference in trends between the mean-rate results and the GSD results. However, if eliminating the LPF simply results in less smoothing of the mean-rate outputs, there might be more complications in the resulting GSD channel outputs. At the computation of GSD, there is a dividing operation of two time-domain waveforms, both of which are now less smoothed The possible benefit of the LPF for GSD processing may be that smoothing helps stabilize the denominator term of the GSD computation by introducing short-term constraints and by suppressing the fast fluctuations in the high-frequency channels.

6.1.3. Elimination of AGC and Synchrony Fall-off

We found that better results were obtained by not using the AGC in many cases, although the performance gain may be marginal. The modified mean-rate front end without the use of the LPF performed just as well as the original full NT version did. On the other hand, the lack of short-term adaptation had a severe negative impact. In order to determine the extent to which short-adaptation is the single dominant component of the NT model after the rectifier, we disabled both the AGC and LPF, leaving in place only the rectifier and short-term adaptation. As in the case of some of the previous results, there were not significant degradation in performance using the CLSTK microphone for testing. However, eliminating both the GSD and LPF produced a significant degradation in recognition accuracy when the CRPZM was used for testing, as well as for the CLSTK microphone when noise was added at an SNR of +20 dB or below.

For these reasons it was concluded that shot-term adaptation by itself is not sufficient to provide good recognition accuracy in adverse conditions.

6.1.4. Amplitude Compression by the Half-wave Rectifier

The NT model includes a compressive nonlinearity that characterizes the halfwave rectification and limited dynamic range of the auditory system. Unlike other non-linear elements such as adaptation and AGC which have major effects on the signal envelope, the amplitude compression is instantaneous and has direct effects on the waveform itself. This reduction of dynamic range, however, may not be necessary or desirable for automatic speech recognition. Also, the existence of the compressive nonlinearity makes the behavior of the whole system dependent on the input amplitude, which complicates the analysis of the front end.

As was described in the previous chapter, the nonlinear compressive characteristics of Seneff's half-wave rectifier is given by the following expression:

$$[n] = \begin{cases} Ghw\left(A \operatorname{atan}\left(Bx[n]\right) + 1\right) & x[n] \geq 0 \\ Ghw\left(e^{ABx[n]}\right) & x[n] < 0 \end{cases}$$

Sullivan and Stern demonstrated that the use of various types of expansive half-wave rectifiers improved the recognition performance in their front end which uses the bandpass filters of the Seneff model followed by half-wave rectification to process the outputs of a multi-sensor array[48]. Sullivan and Stern obtained better performance with these rectifiers than they did with the rectifier used by Seneff.[49]

Motivated by Sullivan's study, we attempted to isolate the contribution of the amplitude-compression elements of the Seneff rectifier, and to determine whether it could be improved or simplified to some degree. We modified the front end by replacing the Seneff rectifier with a linear rectifier with no compression and tested in the AN4 word recognition task. Three alternate versions of linear rectifiers were examined with the input/output characteristics plotted in Figure 6-4 The first version (Lin1) passed only the positive amplitude portion of the input signal. The slope was arbitrarily chosen as unity. For the input amplitude range encountered, Lin1 always produced smaller values than the Seneff rectifier. The second version (Lin3) was designed so that approximately one third of the output signal came had greater amplitude than the output signal from the Seneff rectifier. The slope of the last version (Lin4) was set equal to the initial slope of the Seneff rectifier at the origin, resulting in the output amplitude of Lin4 was always greater than that of Seneff rectifier. In Lin3 and Lin4, a d.c. offset term, *Ghw*, was used for the negative part of input.

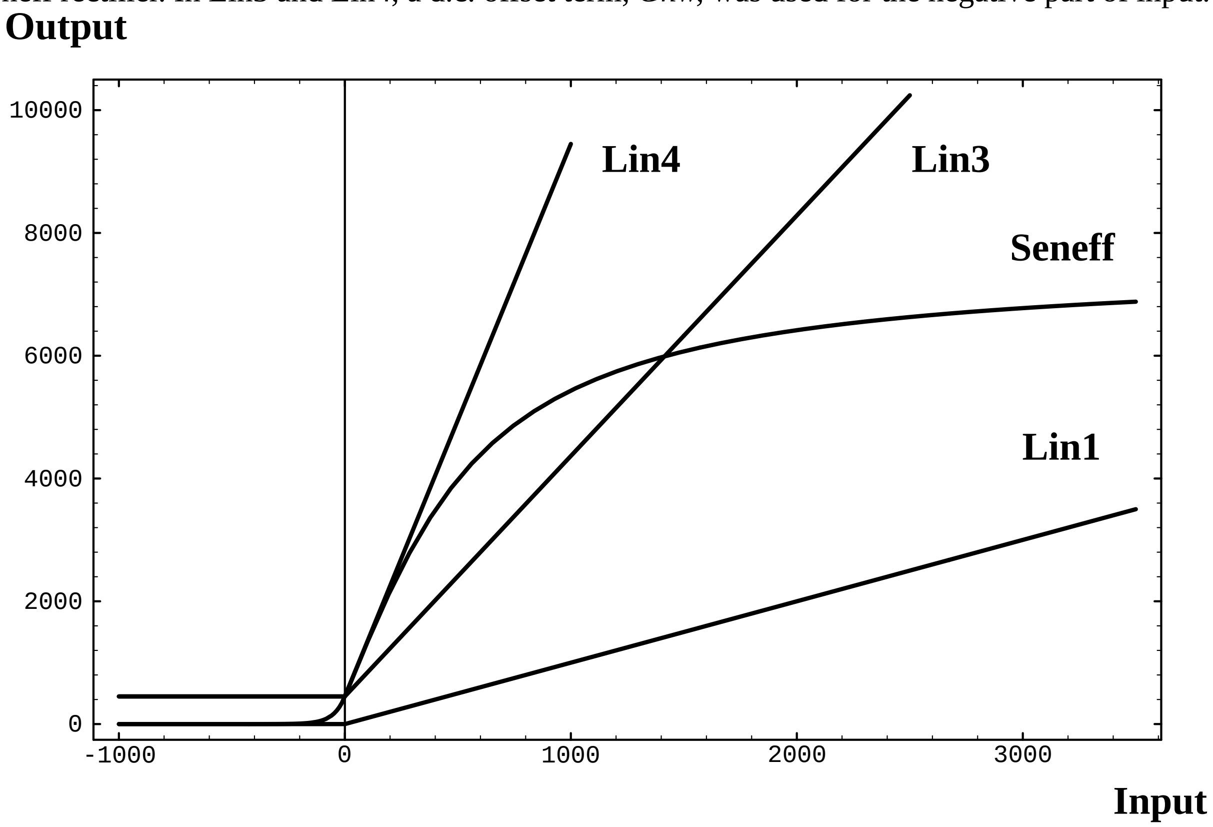

Figure 6-4 I/O characteristics of half-wave rectifiers. Three liner rectifiers are compared with the Seneff rectifier.

The results of these experiments are summarized in Figure 6-5 and Table 6-4 The linear rectifier did not give better performance except for testing using the CRPZM microphone and the mean-rate outputs of the Seneff model. Test results using the CLSTK microphone suggest that the compressive nonlinearity of the Seneff rectifier should be considered to be an integral part of the whole processing, rather than simply preparing the channel signal for later integration to compute short-term energy.

Among the rectifiers tested, Lin3 provided results closest to those of the original compressive Seneff rectifier. Though this is yet to be verified, this is probably because the output range of the processed signal is similar to that of the original. In other words, it is believed that the choice of slope affects the recognition performance due to other nonlinear processing elements in later stages of the NT model. While a more careful search for the best slope of a rectifier like Lin3 may be desirable, we are not encouraged by this experiment.

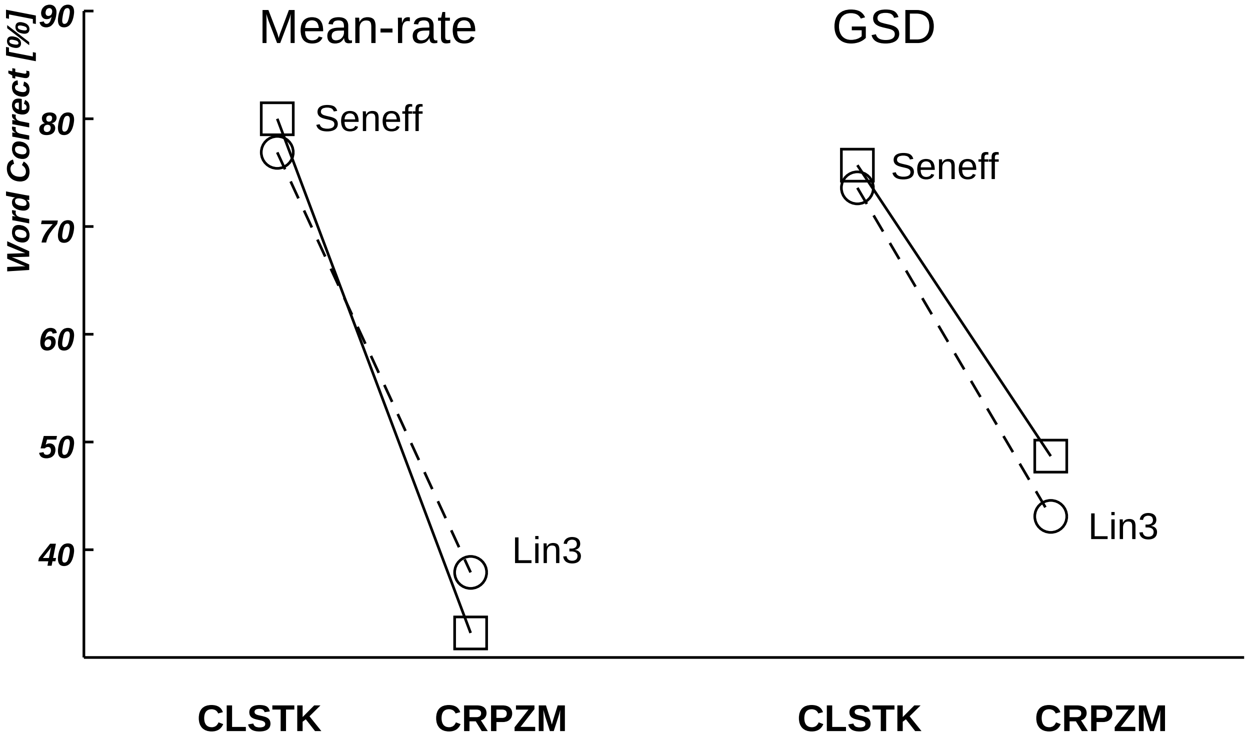

Figure 6-5 Comparison between the Seneff model half-wave rectifier (boxes) and an alternative linear rectifier (circles) for the AN4 task using the mean-rate and GSD outputs of the Seneff model, training with the CLSTK microphone and testing with the CLSTK and CRPZM microphone.

Rectifier Version	Mean-rate outputs		GSD outputs	
	CLSTK	CRPZM	CLSTK	CRPZM
Seneff Rectifier.	80.0%	32.3	75.7	48.7
Linear Rectifier ver. 1	72.0	35.3	70.0	37.2
Linear Rectifier ver. 3	76.9	37.9	73.6	43.1
Linear Rectifier ver. 4	77.2	37.0	72.7	37.7

Table 6-4 Evaluation of several half-wave rectifiers. Seneff’s compressive rectifier was compared with three types of linear half-wave rectifiers.

6.2 Computational Complexity of Implementations of the Seneff Model

Computational models of the auditory periphery almost invariably involve at least band-pass filtering for simulating the processing of the cochlea simulation, and several non-linear stages for NT simulation. Some of these operations are computationally demanding and therefore becomes prohibitive when computing resources are limited. It is of interest to consider execution profile of implemented examples of the Seneff model, in order to understand the relative computational burden imposed by each stage of signal processing.

Two different implementations of Seneff model were used for this study. One was a verbatim implementation by us written in C, which was based on the model described in the published literature[7]. The execution speed for this version is about 43 times real time on a DECstation 5000/200. Its execution profile is itemized by functionality in Table 6-5 About 14% of the total machine cycles are consumed by the BPF stage, about 30% were used by the NT functions, and as much as 56% was used by the computation of the mean-rate and GSD outputs.

Of the amount spent on the NT functions, the rectifier occupied almost one-half of the computation, and the LPF for synchrony fall-off used about one-fourth. The short-term adaptation and the AGC were the least computational burdensome. Internally the short-term adaptation and AGC contain a first order integrator, while the LPF that simulates synchrony fall-off is of the fourth order, and hence more expensive. In any case, there is no doubt that the rectifier is the most costly element among the NT functions because functions from the math library were used to compute the non-linear compression. For the same reason, the GSD costs very much due to computations of its soft limiting functionality.

Another more efficient implementation was developed by the MIT group. In this version, function calls for the rectifier and GSD elements were replaced by table lookup, and consequently a better execution speed of about 17 times real time was achieved. Compared with the above version, it ran 2.534 times faster. The use of table lookup made the GSD processing less computationally demanding, and the relative computational load for the BPF, NT, and GSD stages was 19.87%, 41.64%, and 37.48% respectively.

6.3 Summary

In this chapter we attempted to assess the significance of individual components of the NT stage of the Seneff model. To our disappointment we were unable to identify a single component of the processing that could be eliminated without incurring some loss of robustness in recognition accuracy when speech is presented with distortions due to the effects of additive noise or linear filtering.

Of the various components of the NT model after the rectifier, short-term adaptation appeared to be the most important Seneff model NT processing past the rectifier, and the omission of the AGC

Functionality	%cycles		Turnaround Time Ratio
	Ours	MIT Version	
Cochlea BPF	13.85%	19.87	1.766
Half-wave Rectifier	13.18	13.49	2.476
Short-term Adaptation	3.49	6.88	1.287
Synchrony Fall-off	7.50	12.68	1.500
AGC	5.25	8.60	1.546
Mean-rate	56.23	8.18	3.801
GSD		29.30	
Overall	N/A	N/A	2.534

Table 6-5 Analysis of relative computational complexity of components of the Seneff model, based on the execution profile of CPU cycles. In collecting statistics, a speech segment of 1 second was processed by using program modules tagged by the UNIX `pixie` utility. Also tabulated is a module-by-module comparison of CPU cycles comparison of our implementation of the model with that provided by MIT.

had no significant negative impact. The omission of the LPF to provide synchrony fall-off had little or no significant effect when the CLSTK microphone was used for testing, even with existence additive noise, but it did result in less robustness when the system was tested using the CRPZM microphone. We compared performance using the compressive nonlinearity of the Seneff rectifier with several linear rectifiers and found that the Seneff rectifier provided the best results.

The relative computational cost of Seneff model was roughly 20% for the BPF and 40% each for the NT and the GSD. Among the NT modules, the rectifier and LPF to provide synchrony fall-off were more expensive than the rest.

Chapter 7 Combination of Physiologically-Motivated Signal Processing and Acoustical Pre-processing

In Chapter 5 we noted that the acoustical pre-processing algorithm CDCN produces recognition accuracy equal or better to that of the Seneff model for the AN4 database. In this chapter we describe a series of attempts to combine physiologically-motivated signal processing (using the Seneff model as an example) with acoustical pre-processing algorithms (such as CDCN). The motivation to this work has been to try to achieve further noise robustness by integrating two different techniques that are known to be successful when applied individually.

The ability of physiologically-motivated signal processing algorithms to provide noise robustness has been demonstrated in Chapter 5, and some of the possible reasons underlying the success of these models are explored in Chapter 6. Acoustical pre-processing, in contrast. compensates for environmental degradation by transforming ensembles of cepstral coefficients into those that would be more likely to have been produced by clean speech. While the two approaches to noise robustness may appear to have little in common in their development, they share the goal of achieving noise robustness by providing more reliable feature parameters in different ways.

We considered two different ways of integrating the physiologically-motivated signal processing algorithms with acoustical pre-processing, which are referred to as the waveform-domain approach and the parameter-domain approach, respectively. In one implementation of the waveform-domain approach, the incoming signal is processed by physiologically-motivated front end, and a speech-like waveform is resynthesized from the outputs of the auditory model. The resynthesized waveform is then input to a more conventional front end that includes acoustical preprocessing, like CDCN. In an alternate implementation of the waveform-domain approach, incoming speech is processed using a conventional LPC-based front end with an algorithm like CDCN. A waveform is again resynthesized from the outputs of the of the LPC-based front end, and passed through the auditory model. Using the parameter-domain approach, acoustic “pre-processing” algorithms such as CDCN are applied directly to the outputs of the physiologically-based front end. In our experiments, the raw spectral feature vectors from the Seneff model or their derived cepstral parameters were used.

Recognition experiments were conducted using different training and testing microphones, and with noise added to the speech used for training the system, in order to evaluate the extent to which the combined application of physiologically-motivated signal processing and acoustical pre-processing improves recognition accuracy beyond the levels achieved by either method in isolation.

7.1 The Waveform-domain Approach

In the waveform-domain approach the two approaches (auditory modelling and cepstral pre-processing) are applied in cascade fashion in more or less their original form. Since both types of algorithms normally take speech waveforms as their input, concatenation requires that a speechlike waveform be resynthesized in some fashion.

In principle, the outputs of either acoustical pre-processing and physiologically-motivated processing could be used for the synthesis. One way to accomplish the resynthesis is to extend the LPC analysis/resynthesis paradigm which has been widely used in the area of speech coding, to incorporates the acoustical normalization of cepstral parameters. The other way, which is less well known, is to reconstruct the wide-band waveform from the narrow-band channel waveforms resulting from physiologically-motivated processing.

In the following subsections, we discuss these two techniques of combining acoustical pre-processing and physiologically-motivated signal processing in the waveform domain.

7.1.1. Waveform Resynthesis from LPC-derived Cepstra Followed by the Seneff Model

Perhaps the most obvious approach to model combination is resynthesis of the waveform from cepstral coefficients. The acoustical pre-processing methods proposed by Acero[2] all use ensembles of cepstral coefficients as input and output. The LPC cepstral parameters c_n are directly related to the original LPC parameters a_n by the recursive conversion formula below:

$$c_i = \begin{cases} ln(G) & ;i=0 \\ a_1 & ;i=1 \\ a_i + \sum_{j=1}^{i-1} \frac{i-j}{i} c_{i-j} a_j & ;1 < i \le p \\ \sum_{j=1}^{p} \frac{i-j}{i} c_{i-j} a_j & ;i > p \end{cases}$$

The inverse relationship, which is described in Appendix A [10] is

$$a_i = \begin{cases} c_1 & ;i=1 \\ c_i - \sum_{j=1}^{i-1} \frac{j}{i} a_{i-j} c_j & ;1 < i \le p \end{cases}$$

We extend the ordinary LPC analysis/resynthesis so that speech can be resynthesized from sequences of cepstral coefficients that had been normalized by CDCN or some other normalization algorithm such as SDCN. The resynthesized speech waveform is then processed by the Seneff model in the usual fashion to produce the pseudospectral parameters as the feature vectors for the recognition system.

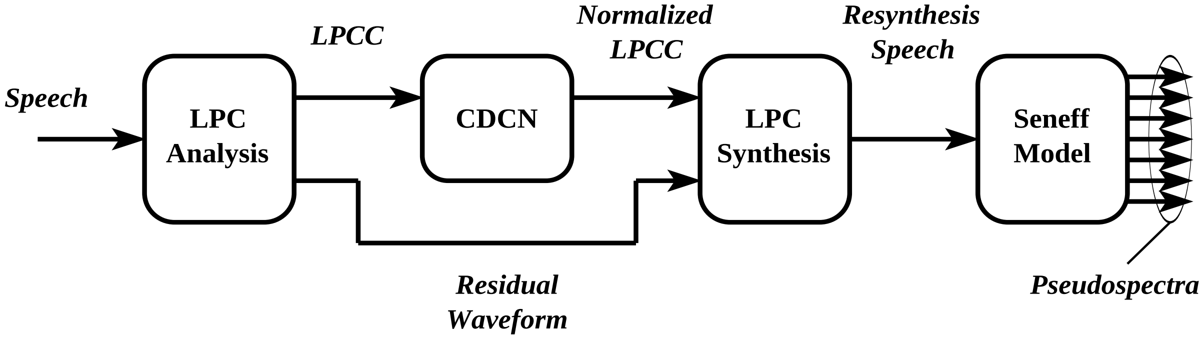

Figure 7-1 The waveform-domain integration of acoustical pre-processing and physiologically-motivated signal processing. The speech waveform is first input to a modified LPC vocoder, with a synthesis filter that reflects the effects of CDCN. The resynthesized speech is then input to the Seneff model to produce the mean-rate and GSD pseudospectra as feature parameters for the recognizer.

The block diagram shown in Figure 7-1 describes the waveform-domain integration of LPC analysis/synthesis with CDCN and the Seneff model. In the analysis phase, the original, 12th-order LPCC parameters were used to build the LPC analysis filter based on the input speech. The LPC analysis filter parameters were updated at every sample point to produce the residual error in a seamless manner. Interpolation of the filter parameters was done in the cepstral domain after undoing the effect of frequency warping, which was used by the SPHINX front-end and the CDCN training and normalization routines. For an N-sample point utterance analyzed using an analysis frame of length L updated by U points, let the center of the k-th analysis frame

$$n = kU + \frac{L}{2} \qquad ;k = 0, 1, 2, \ldots$$

be representative of the quasi-stationary system described by a set of LPCC parameters:

$$c_i(k) \qquad ;i = 1, 2, \ldots 12$$

The linearly interpolated LPCC between the k^{th} and the $(k+1)^{\text{th}}$ frames at time n are:

$$c_i\,[n] = \frac{(\,(k+1)\,U + \frac{L}{n} - n)\,c_i(k) + \,(n - kU - \frac{L}{2})\,c_i(k+1)}{U} \qquad ;i = 1, 2, \ldots 12$$

which are transformed into LPC parameters as follows:

$$a_i[n] = \begin{cases} c_1[n] & ;i=1 \\ c_i[n] - \sum_{j=1}^{i-1} \frac{j}{i} a_{i-j}[n]\, c_i[n] & ;1 < i \le 12 \end{cases}$$

The input speech was made to be zero mean and passed through the first order HPF:

$$H_{HPF}(z) = 1 - 0.97z^{-1}$$

which matches the pre-emphasis of the LPC analysis at the SPHINX front-end. The resulting error signal $e[n]$ from the LPC analysis filter was preserved for later use as the excitation function in the resynthesis project. For zero-mean input speech $s[n]$ and the LPC analysis filter characterized by the interpolated LPC parameters shown above:

$$x[n] = s[n] - 0.97s[n-1]$$
$$e[n] = x[n] - \sum_{i=1}^{12} a_i[n]\, x[n-i]$$

The LPCC coefficients used for the analysis were normalized by CDCN. The CDCN training for the target environment (CLSTK) had been provided externally off line.

The synthesis was the inverse of the analysis, except for the use of LPCC normalized by CDCN:

$$c'_i(k) \qquad ;i = 1, 2, \ldots 12$$

The transfer function of the LPC synthesis filter for the implied spectral envelope was computed from the cepstral coefficients after CDCN. The synthesis filter was excited by the residual error signal $e[n]$ that had been saved in the LPC analysis procedure, and finally the effect of preemphasis was compensated by the LPF:

$$H_{LPF}(z) = \frac{1}{1 - 0.97z^{-1}}$$

to produce the resynthesized speech $r[n]$:

$$y[n] = e[n] + \sum_{i=1}^{12} a'_i[n] y[n-i]$$
$$r[n] = y[n] + 0.97r[n-1]$$

The normalized LPCC parameters occasionally resulted in an unstable synthesis filter. As a remedy, the magnitude of reflection coefficients was checked at every sample point and poles located outside the unit circle were reflected inside to regain the stability. This procedure was equivalent to passing the resynthesis waveform through an appropriate all-pass filter whenever instability was detected.

The numerical accuracy of the reconstructed filter parameters is affected by the truncation of the cepstral coefficients to a finite order that occurs n normal LPC analysis. The problem of numerical accuracy arises especially during the conversion of LPC parameters to and from LPC cepstra, and in the application of the bilinear transform between the cepstral parameters to provide frequency warping. The maximum amount of observed numerical error for LPC coefficients was about 10%. By informal listening to processed speech, we confirmed that the CRPZM speech with muffled low-pass characteristics was cleaned up. The processed CRPZM speech was subjectively very similar to that of the processed CLSTK speech.

Despite the lack of audible degradation in the resynthesized speech, the loss of information associated with the analysis/resynthesis may affect speech recognition accuracy. We first examined this possibility with a preliminary experiment that omitted the CDCN stage. Recognition accuracy was evaluated for the resynthesized speech from the AN4 corpus using SPHINX and the standard LPCC front end. The system had been trained on the CLSTK microphone data for this baseline experiment.

Table 7-1 shows the results of this preliminary experiment. Although the difference is small, the 3.6% drop in recognition accuracy using the CLSTK microphone indicates that there is a loss of fidelity in the resynthesis. It is not likely that the CRPZM results are statistically significant. For the combined approach to be useful, CDCN must provide enough of an improvement to overcome this 3.6% “handicap”.

Figure 7-2, Figure 7-3, Table 7-2 and Table 7-3 summarize the results a series of experiments evaluating the CDCN-based resynthesis technique. These experiments compare recognition accuracy for resynthesized speech input to the Seneff model. We describe results obtained with both the mean-rate and synchrony outputs of the model, and the conventional front end that uses LPC cepstra. Results are shown both with CDCN and without CDCN. The CLSTK microphone was used for training, and either the CLSTK or CRPZM microphone was used for testing, as indicated.

Waveform	CLSTK Testing	CRPZM Testing
Original	85.3%	18.6
Resynthesis	81.7	19.8

Table 7-1 Preliminary experimental results for AN4 word recognition accuracy using the LPCC front end, testing on the original and the resynthesized speech waveforms obtained from the CLSTK and CRPZM microphones. The input speech waveform underwent the LPC analysis/resynthesis based on the LPCC parameters restored after the bilinear transform.

As can be seen from Figure 7-3, the combination with CDCN improves the performance of both the mean-rate and synchrony outputs of the Seneff model in the “cross condition”, training with the CLSTK and testing with the CRPZM microphone. The performance obtained using the mean-rate outputs showed a 18.5% improvement, while the use of the GSD outputs produced an improvement of only 3.9%. The GSD-based front end is less affected by changing the acoustical environment and it outperforms the mean-rate front end in the baseline condition. Since CDCN-based resynthesis is believed to cause the spectral envelope of the output signal to resemble more closely that of the target environment (CLSTK microphone, it is reasonable to observe more obvious improvement with the mean-rate front-end than with the GSD front-end.

Front-End	CLSTK	
	CDCN Resynthesis	Baseline
Mean-Rate	70.6%	80.0
GSD	70.6	75.7
LPCC	74.9	85.3

Table 7-2 AN4 word recognition rates obtained using resynthesized speech after CDCN processing, using the mean-rate and GSD outputs of the Seneff model. Result using the LPCC front end are also shown for comparison. The CLSTK speech was processed by the CDCN algorithm, resynthesized, and input to the Seneff model. The CDCN training was done using the CLSTK microphone with LPCC.

It should be noted, however, that recognition accuracy using this approach is still substantially worse than what is obtained with LPC cepstra and CDCN in its original form. The direct application of CDCN to LPCC parameters improved the CRPZM result up to about 75%, while the attempt to normalize the noisy speech using the waveform-domain approach brings it to 31.7%.

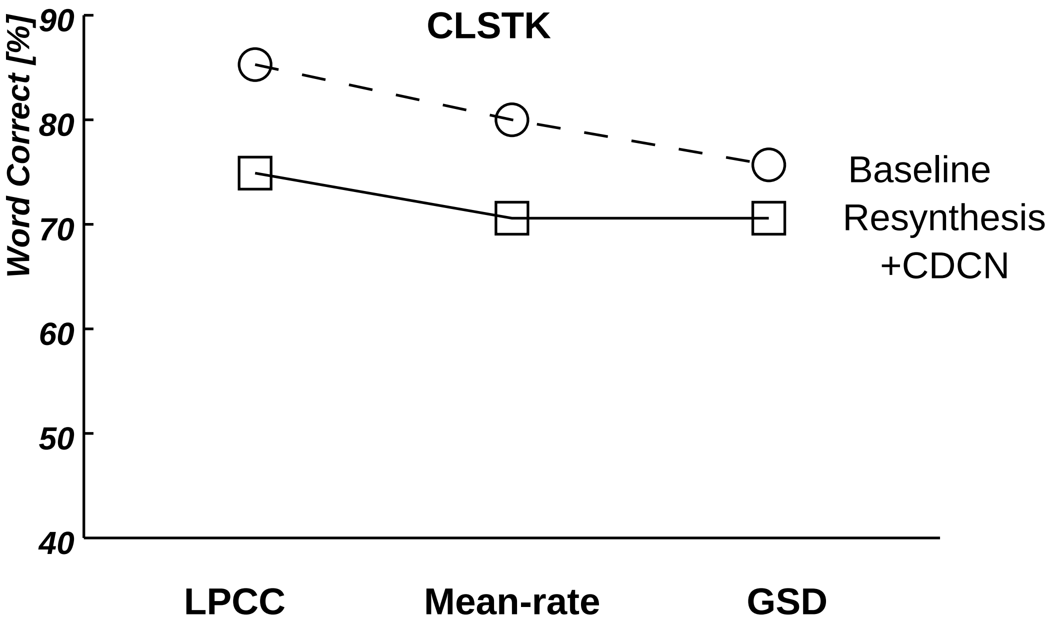

Figure 7-2 AN4 word recognition rates obtained using resynthesized speech after CDCN processing, using the mean-rate and GSD outputs of the Seneff model. Result using the LPCC front end are also shown for comparison. CLSTK speech was processed by the CDCN algorithm, resynthesized, and input to the Seneff model. The CDCN training was done by using the CLSTK microphone with LPCC. Broken lines in the graph indicate the corresponding baseline performance of unprocessed CLSTK speech.

It should also be noted that the waveform resynthesis based on the CDCN degrades recognition performance for the clean test data using the CLSTK microphone. This would be accounted for by the following argument. The CDCN algorithm produces an ensemble of cepstrum parameters, which optimizes the partition of the parameter space based on the concept of a universal codebook, and the outcome of codebook generation is different from that of the original LPCC. Likewise, the CDCN-based resynthesis enhances the noisy speech and transforms it into the domain associated with the universal codebook. However, this system was trained by using the feature vectors obtained from clean training data, which is expected to be very similar to the universal codebook but not optimal. There is room for further improvement by processing both training and testing speech using CDCN-based resynthesis.

Front-End	CRPZM	
	CDCN Resynthesis	Baseline
Mean-Rate	50.8%	32.3
GSD	52.8	48.7
LPCC	31.7	18.6

Table 7-3 AN4 word recognition rates obtained using resynthesized speech after CDCN processing, using the mean rate and GSD outputs of the Seneff model. Result using the LPCC front end are also shown for comparison. The CRPZM speech was processed by the CDCN algorithm, resynthesized, and input to the Seneff model. The CDCN training was done using the CLSTK microphone with LPCC.

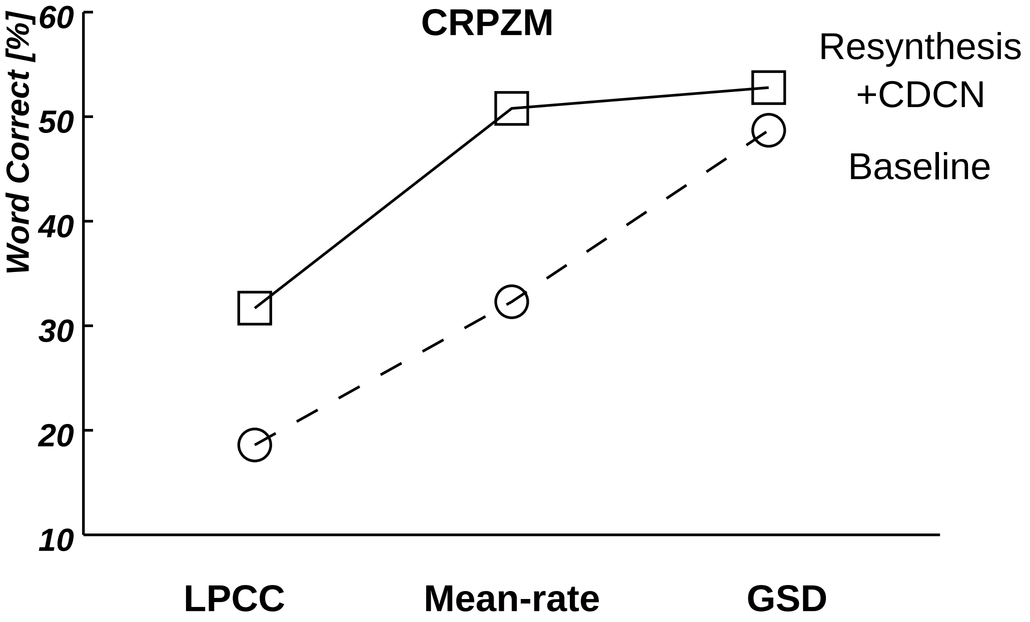

Figure 7-3 AN4 word recognition rates obtained using resynthesized speech after CDCN processing, using the mean-rate and GSD outputs of the Seneff model. Result using the LPCC front end are also shown for comparison. CRPZM speech was processed by the CDCN algorithm, resynthesized, and input to the Seneff model. The CDCN training was done by using the CLSTK microphone with LPCC. Broken lines in the graph indicate the corresponding baseline recognition accuracy produced by unprocessed CRPZM speech.

In summary, we found that the CDCN algorithm to be very effective when used to normalize the LPC resynthesis filter. Speech recognition performance by the mean-rate and GSD front-end was improved in the noisy test environment by using the CDCN resynthesis.

7.1.2. Waveform Reconstruction from the Outputs of the NT Model Followed by CDCN

It is also possible to reconstruct wide-band speech waveform by using the outputs of the NT model of the physiologically-motivated front end. The channel outputs from the NT model not only reflect the advantages in noise robustness attributed to non-linear processing, but also are based on the narrow-band analysis of the cochlear BPF, which is expected to preserve relevant phonetic information. Given that assumption, if one could band-limit the signal in each channel (which was originally narrowband but now exhibits greater spectral spread after the nonlinearity, the sum of such post-filtered channel signals would form a useful wide-band waveform. The reconstructed speech can then be subjected to acoustical "pre-processing" for further noise robustness.

The key element of this strategy is to build a good reconstruction filter and to design a summing principle. This is efficiently realized by "backward filtering" of the Seneff BPF bank that models the cochlea filters. In the next few paragraphs we describe the implementation of the reconstruction BPF and how information from the various NT channels is combined.

7.1.2.1. Implementation Details of the Technique

Figure 7-4 shows the block diagram of the signal processing used for this procedure, including the Seneff BPF, the NT processing, and the reconstruction filter bank based on the Seneff BPF. In each BPF channel, the cascade/parallel implementation of the Seneff BPF has the overall transfer function

$$H_i(z) = \sum_{n=-\infty}^{+\infty} h_i[n]\, z^{-n}$$

$$= \frac{G_i\,(1 + a_{zero_i} z^{-1} + b_{zero_i} z^{-2})^2 \prod_{j=1}^{i} (1 + a_{cascade_j} z^{-1} + b_{cascade_j} z^{-2})}{(1 + a_{pole_i} z^{-1} + b_{pole_i} z^{-2})^2}$$

$$\times \prod_{k=1}^{4} (1 + a_{initial_k} z^{-1} + b_{initial_k} z^{-2}) \qquad ; i = 1, 2, \ldots, 40$$

and the channel output signal from the BPF i to the input speech $x[n]$ is

$$y_i[n] = h_i[n] \otimes x[n]$$

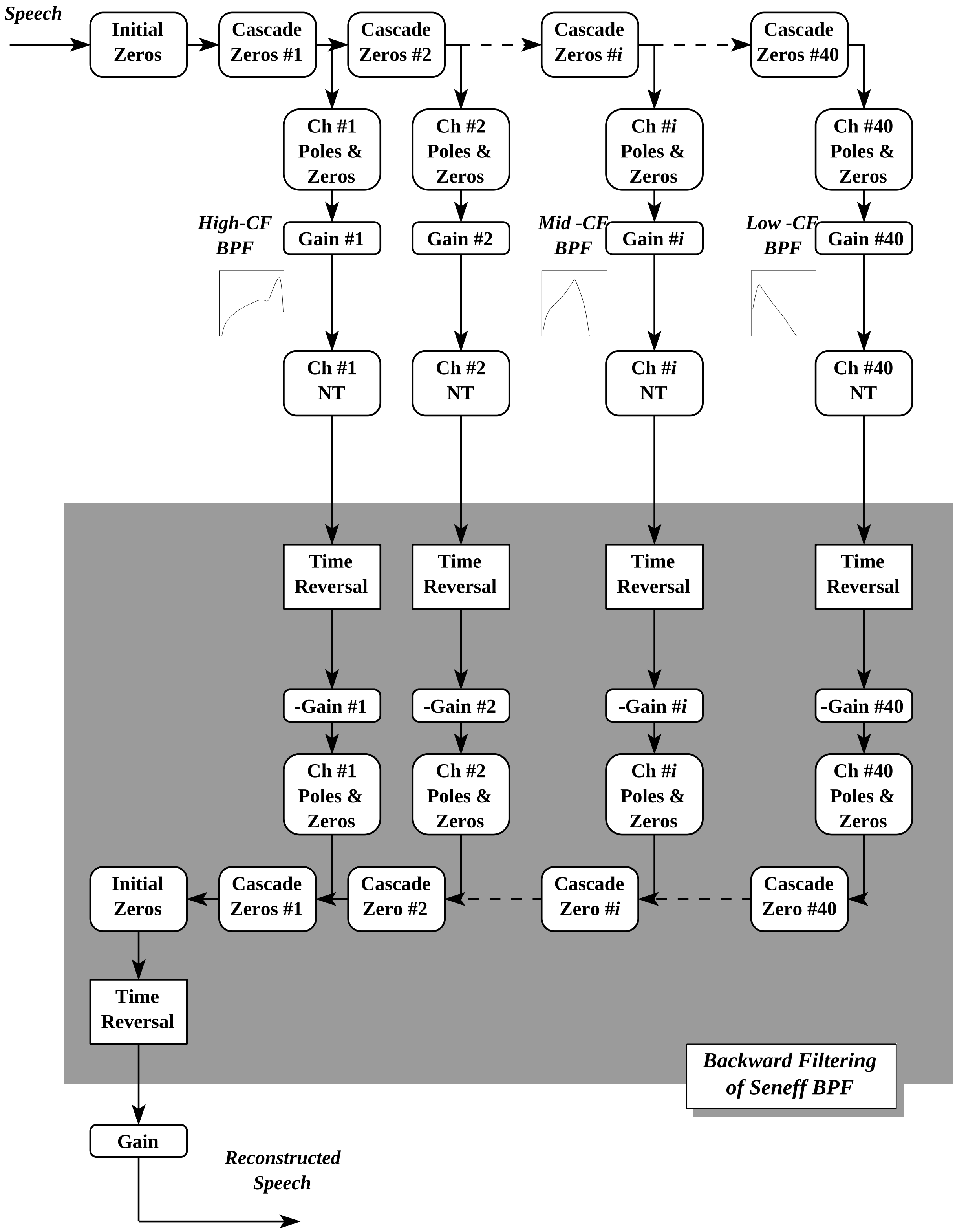

Figure 7-4 Reconstruction of wideband speech by backward filtering of the NT-channel waveforms using the Seneff BPFs. The reconstructed output waveform was input to the SPHINX LPCC front-end for further environmental compensation using the CDCN algorithm.

As we studied in Chapter 4, the NT stage following the BPF includes nonlinear signal processing. For the sake of brevity, we denote the input/output relationship at the channel i NT processing as follows

$$v_i[n] = NT_i[y_i[n]]$$

Each channel signal is reversed in time, compensated in gain, and passed through the corresponding channel filter in the parallel branch. The branching points of the original cascade/parallel chain now become the summing points of the filtered channel signals and the partially reconstructed waveform in the cascade chain. The transfer function of the backward BPF is the same as the first stage except for the gain normalization term

$$F_i(z) = \sum_{n=-\infty}^{+\infty} f_i[n]z^{-n}$$

$$= \frac{G'_i(1+a_{zero_i}z^{-1}+b_{zero_i}z^{-2})^2\prod_{j=1}^{i}(1+a_{cascade_j}z^{-1}+b_{cascade_j}z^{-2})}{(1+a_{pole_i}z^{-1}+b_{pole_i}z^{-2})^2}$$

$$\times\prod_{k=1}^{4}(1+a_{initial_k}z^{-1}+b_{initial_k}z^{-2}) \qquad ;i = 1, 2, \ldots, 40$$

Due to the nonlinearity of the NT processing, it is not possible to define the overall gain of the system for each channel. However, the NT modules are set so that the output signal level is balanced to some extent among channels and seems to maintain the magnitude relationships between frequency bands observed at the BPF. The BPF gains G_i are defined so that each channel output at its characteristic frequency is approximately unity gain with a slight high-frequency preemphasis. Therefore, the backward BPF gains G'_i are set such that the unity amplitude response is realized throughout the linear filtering part of the channel at its characteristic frequency, *i.e.*

$$\left|H_i(e^{j\omega})F_i(e^{j\omega})\right|\Big|_{\omega=2\pi CF_i} = 1$$

After undergoing the cascade chain and the initial zeroes of the backward BPF, the resulting wideband waveform is the sum of individual channel signals

$$r[n] = \sum_{i=1}^{40} f_i[n]\otimes v_i[-n]$$

The signal waveform is reversed in time again, and the amplitude is normalized so that the total energy of the reconstructed signal equals that of the input speech. The reconstructed wide-band speech is obtained from the relations

$$s[n] = Ar[-n]$$

$$A = \frac{\overline{x[n]}}{\overline{r[n]}}$$

An additional advantage with this configuration is that the use of backward filtering preserves phase alignment among the channel signals.

7.1.2.2. Experimental Results

The analysis/synthesis procedure we have described may still alter the frequency response because of frequency components that fall between the center frequencies of the channel filters. The overall magnitude response of the analysis and reconstruction filter banks are not uniform over the whole frequency range. It was concluded that the impact of this problem was insignificant by evaluating recognition accuracy for resynthesized speech obtained by cascading forward and backward filtering back-to-back. The result of this preliminary experiment is summarized in Table 7-4, and they indicate that the effect of cascaded BPFs on the overall frequency response was negligible. Furthermore, for our particular application, the impact of this problem should be ameliorated to some degree because of the nonlinearity that spreads out the channel signal spectra. This experiment was implemented by combining Seneff’s rectifier, short-term adaptation, and AGC, but excluded the LPF, which smooths out high-frequency components of the channel signal. We regard detailed waveform information such as level crossing information to be important as it is known to carry the message contents.

Waveform	CLSTK Testing	CRPZM Testing
Unprocessed (Baseline)	85.3%	18.6
BPF + Backward BPF	84.2	18.1

Table 7-4 Results of a study to demonstrate the feasibility of the reconstructing the speech waveform by backward filtering through the bank of cochlear BPFs. Recognition accuracy is described for the AN4 task using the LPCC front end. Reconstructed speech using the CLSTK and CRPZM microphones were compared to the original speech. The input speech was first analyzed using the Seneff BPF and subsequently reconstructed by summing the results of backward filtering by the cochlear over all channels

Table 7-5 summarizes the results of AN4 testing based speech that is reconstructed using the standard LPC cepstra with and without CDCN. We note that these results are very similar to the baseline recognition rates using LPCC, (85.3% for the CLSTK data and 18.6% for the CRPZM data). With CDCN processing, the results are also very similar to what we already observed with the baseline LPCC with CDCN. It was suggested that the NT modules do not improve noise robustness in this configuration.

Even though our synthesis of speech from the NT information did not produce improved recognition accuracy, we gained a great deal of useful insight from informal listening to reconstructed speech. First, the reconstructed speech was intelligible enough to human ears after the nonlinear transformation of channel signals by the NT modules. The relevant information was preserved in the NT channel signals and was successfully reconstructed. We are encouraged to consider this type of processing for speech enhancement in noise. Second, the NT processing produced audible effects in the reconstructed speech: the dynamic range of the speech was narrowed and the noise floor was higher than the original relative to the message. This effect is not a desirable aspect of NT processing if the reconstructed speech is to serve human listeners.

Front-End Version	Language Weight	CLSTK Testing	CRPZM Testing
LPCC Baseline of Reconstructed Speech	4.0	84.8%	18.0
	7.0	83.1	29.5
With CDCN	5.5	82.1	72.8

Table 7-5 AN4 word recognition performance by the LPCC front-end with and without CDCN. Tested were the original and the reconstructed speech waveforms using the CLSTK and CRPZM microphones. The input speech waveform was processed by the rectifier, adaptation, and AGC NT modules of the Seneff model and was reconstructed by summing the results of backward filtering by the cochlear BPFs over all channels.

7.2 Parameter-domain Approach

In Section 7.1, we discussed two techniques to combine physiologically-motivated signal processing with acoustical pre-processing in the waveform domain. As an alternative approach to such waveform-domain procedures, we also attempted to integrate them in the feature parameter domain. In this approach, speech is first transformed into spectral or cepstral parameters by the physiologically-motivated signal processing and then processed directly by acoustical pre-processing algorithms (without recovering a waveform as an intermediate step).

The channel responses from the physiologically-motivated signal processor, when averaged and down-sampled in time and then aligned channel by channel along the frequency axis, could be interpreted as a short-term energy representation of speech. We this representation the *pseudospectrum* of speech. In the baseline experiments and in the evaluation of functional components of the NT model, we used these pseudospectra based on the outputs of mean-rate or GSD processing as

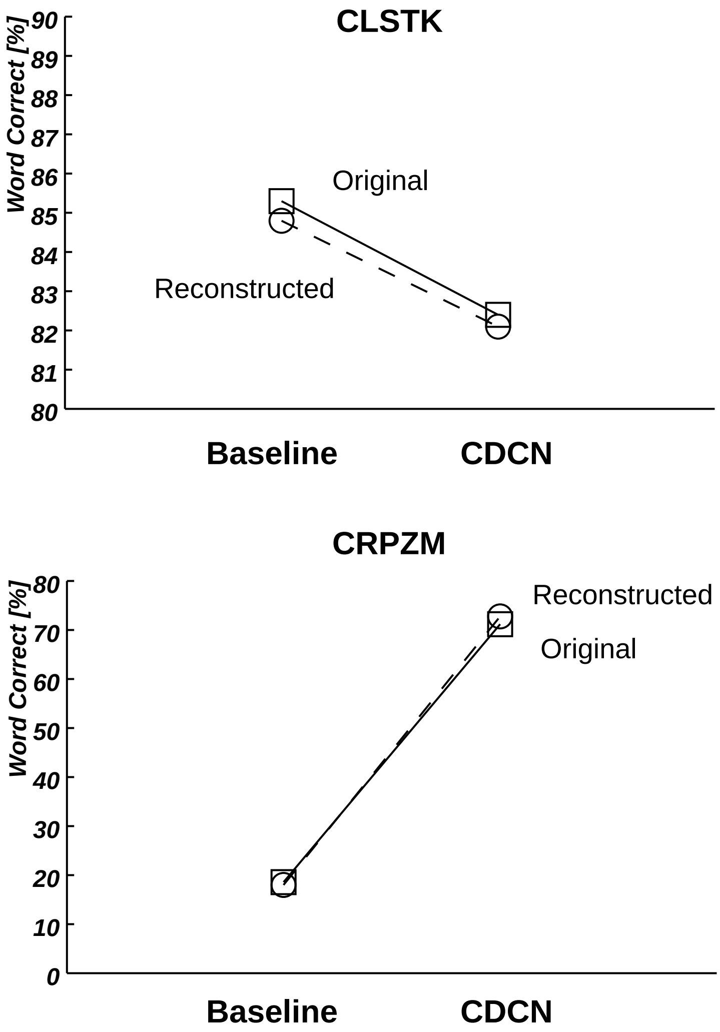

Figure 7-5 AN4 word recognition performance by the LPCC front-end with and without CDCN. Tested were the original and the reconstructed speech waveforms using the CLSTK and CRPZM microphones. The input speech waveform was processed by the rectifier, adaptation, and AGC NT modules of the Seneff model and was reconstructed by summing the results of backward filtering by the cochlear BPFs over all channels.

the raw feature parameters. In the parameter-domain approach, these spectrum-like features are used for spectral normalization or further converted into the cepstral domain (and hence called the pseudocepstra) for the use of successful cepstral normalization techniques known as acoustical pre-processing.

In the following subsections we first discuss SNR-dependent spectral normalization of the pseudospectra (SDSN). We then describe cepstral normalization applied to the pseudocepstra derived from the mean-rate and the GSD output parameters by using all-pole modelling or inverse cosine transform.

7.2.1. SNR-Dependent Normalization of Mean-rate Pseudospectra

In the first attempt to implement the parameter-domain approach, we applied an SNR-dependent normalization technique to the mean-rate pseudospectra[50]. We used the normalization procedure originally proposed by Acero as SDCN, and applied it to the spectral domain (and therefore it is referred to as SDSN). This technique is particularly attractive because the normalization procedure is extremely efficient, and the training procedure is data driven.

SDCN provides an additive correction vector in the cepstral domain to provide joint compensation for the effects of additive noise and linear filtering. The specific correction vector chosen depends on the SNR, which is measured on a frame-by-frame basis. Additive correction vectors in the cepstral domain or log-spectral domain directly compensate for the effects of linear filtering, but they also can compensate for the effects of additive noise as well (on an incrementally-linear basis)[51]. The mean-rate pseudospectra are in effect a compressed characterization of speech spectra. Therefore, since the SDCN algorithm worked well for the cepstral parameters is expected to work similarly well for the mean-rate parameters.

We now describe the training procedure for the SNR-dependent normalization. Two datasets representing speech from the CLSTK and CRPZM microphones were processed by the Seneff model. For each dataset, parameters from the mean-rate outputs were collected for every frame, and sorted by instantaneous SNR. Average cepstra were computed for each SNR. 30 bins were needed to cover the dynamic range of the AN4 speech data at the resolution of 1 dB/bin.

During the normalization process, the SNR of the current frame is estimated, and a correction vector is selected that is appropriate for that SNR. Normalization is achieved by adding the correction vector to the noisy mean-rate spectrum of the current frame.

Figure 7-6 and Figure 7-7 show the results of initial attempts to implement the parameter-domain approach. The SNR-dependent normalization algorithm was applied in the spectral domain to normalize pseudospectra from the mean-rate outputs of the Seneff model. In the first experiment (Figure 7-6), the compensation vectors were estimated to compensate for CRPZM data. The use of SDSN improved the word recognition rate produced by the physiologically-motivated front-end from 32.3% to 64.0%. Previously it had been shown that the use of SDCN enhanced the recognition rate obtained using LPCC from 18.6% to 67.2%.

In the second experiment (Figure 7-7), the normalization parameters were estimated in the presence of white noise added at a 20 dB global SNR, and applied to the AN4 testing set corrupted by noise at 10, 20, and 30 dB global SNR. The use of physiologically-motivated processing in conjunction with SDSN provides recognition accuracy that is clearly better than the accuracy obtained using conventional LPC-cepstral processing.

To summarize, the SDCN algorithm was successfully applied to mean-rate parameters and significantly improved recognition accuracy. The recognition rate achieved using this processing was significantly higher than that of the baseline LPCC with SDCN with additive noise, and was comparable to the baseline obtained for the CRPZM speech with no noise.

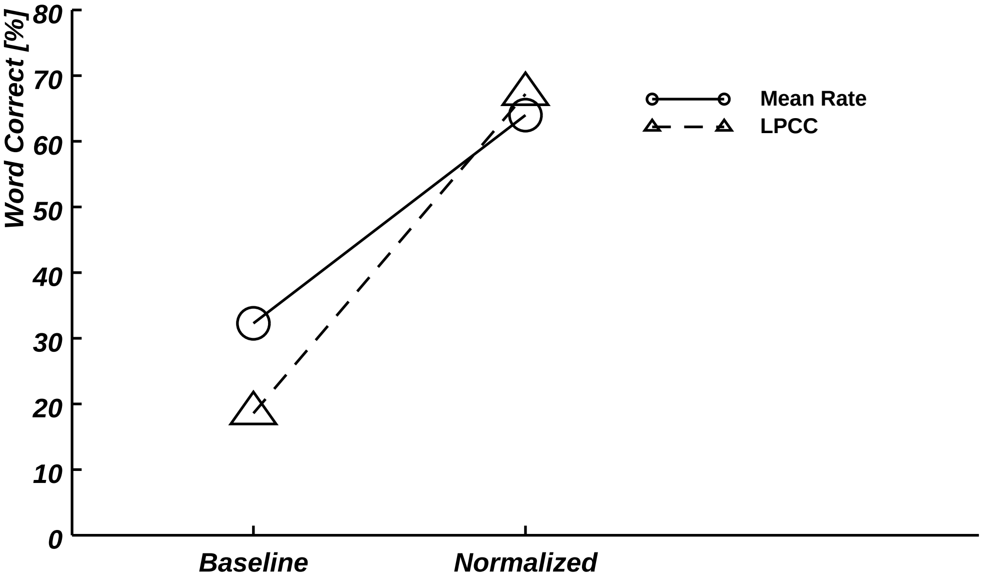

Figure 7-6 Improvement in recognition accuracy obtained using the parameter-domain approach. Circular symbols represent the AN4 word recognition rates using the mean-rate spectra of the Seneff model with and without additional spectral normalization by SDSN (labelled Baseline and Normalized respectively). Triangular symbols represent performance using the LPCC front-end with and without SDCN

7.2.2. AP-derived LPC Cepstra used in Combination with CDCN

It is also of interest to apply CDCN to feature vectors derived from physiologically-motivated processing (referred as “AP-derived” hereafter). This could be realized by converting the raw feature vectors output from the physiologically-motivated processing into cepstrum-like forms. The transformation from pseudo-log-spectrum to cepstrum could be accomplished either by use of LPC-type all-pole modelling or more directly by use of the discrete cosine transform (DCT) of the NT spectra. In this section and the one that follows, our procedures to extract cepstral information and its combination with CDCN are described.

One could compute a set of parameters that are similar in nature to cepstral parameters but that are derived from the output of Seneff model. The block diagram shown in Figure 7-8 describes the procedure to extract cepstrum-like parameters based upon such pseudospectra. The output of the Seneff model is downsampled and expanded in magnitude, if appropriate, and it undergoes the inverse discrete cosine transform. The outcome is a sequence in the time domain that could be interpreted as an pseudoautocorrelation sequence based on the Seneff model output. The best-fit all-

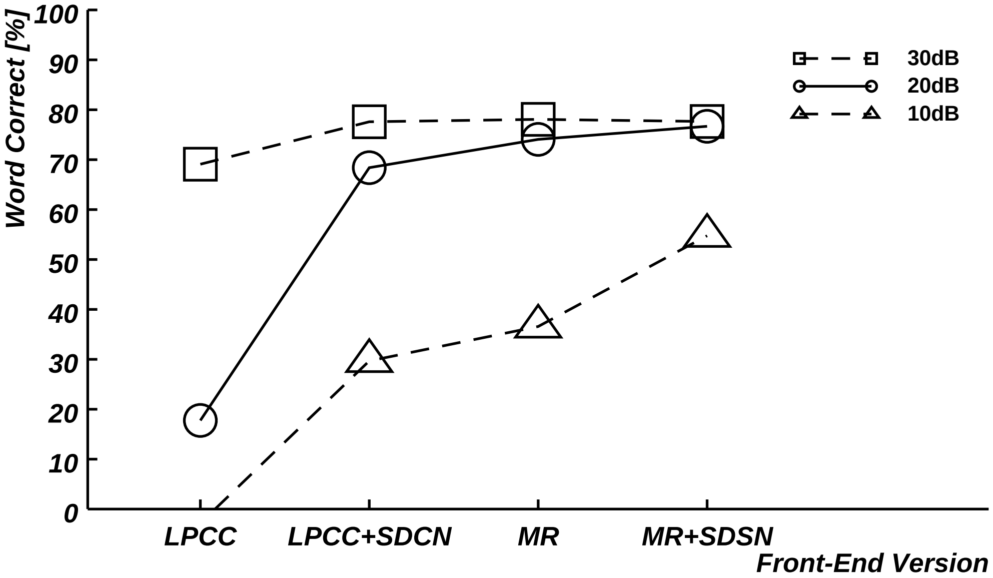

Figure 7-7 Performance improvement using the parameter-domain approach under additive noise conditions. AN4 word recognition rates using the mean-rate outputs of the Seneff model are compared to results using LPCC parameters. Results are shown for speech from the CLSTK microphone with additive white noise at 10, 20, and 30 dB global SNR with and without the use of SNR-dependent parameter normalization. The SDSN and SDCN compensation parameters were trained using speech from the CLSTK microphone with noise added at a global SNR of 20 dB

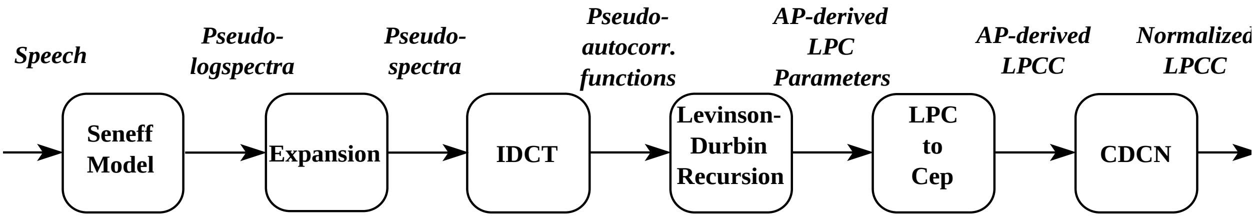

Figure 7-8 Derivation of AP-derived cepstral parameters for the parameter-domain integration of the Seneff model and CDCN. Pseudospectra from the Seneff model are expanded, if necessary, to compensate for the compressive nature of the model. The resulting spectra are fitted by all-pole modelling and the LPCC parameters are computed.

pole model of the pseudospectra is obtained by applying the Levinson-Durbin recursion[52] to the sequence.

Two issues of concern that must be addressed are the lack of information in the Seneff model outputs about D.C. components and high-frequency components of the original speech sounds, and the non-uniform frequency sampling of the Seneff model. One possible solution to these problems is the use of the frequency-selective LPC technique[9] which enables us to fit an all-pole model only to part of the spectrum instead of to the entire bandwidth. Non-uniformity along the frequency axis is in fact advantageous for our purpose. It is reasonable to describe the discrete energy spectra of speech sampled along a frequency scale such as the Bark scale, in terms of the LPC parameters.

The resulting LPC parameters are converted into cepstral coefficients by a well-known recursion formula[10] with which SPHINX front-end had been optimized.

The procedure to compute the AP-derived LPCC is the following. Let

$$S[k] \qquad ;k = 1, 2, \ldots, 40$$

denote the downsampled channel outputs of the Seneff model aligned in ascending order in frequency. Because the coefficients $S[k]$ can be considered to be a discrete representation of the compressed energy spectrum of the input speech, one can restore prominent spectral peaks by using an expansive function, in order to prepare for fitting the parameters of the all-pole model

$$S'[k] = EXP(S[k])$$

A pseudoautocorrelation sequence r_i is obtained by computing the inverse cosine transform of $S'[k]$

$$r_i = \sum_{k=1}^{40} S'[k] \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{40}\right] \qquad ;i = 1, 2, \ldots, p$$

Given the p-th order LPC model

$$H(z) = \frac{G}{A(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}}$$

the LPC cepstral parameters are computed by the procedure described in Appendix A as follows:

$$c_i = \begin{cases} ln(G) & ;i{=}0 \\ a_1 & ;i{=}1 \\ a_i + \sum_{j=1}^{i-1} \frac{i-j}{i} c_{i-j} a_j & ;1 < i \le p \\ \sum_{j=1}^{p} \frac{i-j}{i} c_{i-j} a_j & ;i > p \end{cases}$$

We note that there is an associated issue with the use of the derived energy term. In the 3-codebook discrete HMM version of SPHINX used, the short-term energy information, in terms of c_0, of each frame was used as well as the cepstral and differential cepstral parameters. This energy information in its original form was from the straightforward LPCC parameters obtained from the speech, and it is still available by directly computing the sum of squared sample values of the windowed input waveform. The other definition of c_0 here is, as is given in the formula, the logarithm of the predictor gain of the AP-derived LPCC, which should not be confused with the ordinary LPCC.

We conducted a pilot experiment in which we compared AP-derived LPCC parameters based on the mean-rate and the GSD feature vectors, to the original feature parameters with and without features representing short-term energy. This was studied by replacing the original energy estimate with the AP-derived one. The results are shown in Table 7-6 and it was clear that the AP-derived energy estimate is superior to the original for all test conditions.

Processor Version	Language Weight	CLSTK Testing	CRPZM Testing
GSD-derived LPCC w/ Raw Frame Energy	5.5	77.8	51.3
	8.0	76.6	53.6
GSD-derived LPCC	5.5	75.3	60.8
	6.5	75.8	60.8
MR-derived LPCC w/ Raw Frame Energy	5.5	82.0	28.7
	8.5	79.8	34.2
MR-derived LPCC	5.5	82.4	34.7
	9.5	79.3	39.5

Table 7-6 Effects of c_0 on the AN4 task. Alternative formulations of c_0 derived from the all-pole modelling of the mean-rate and GSD outputs are compared with c_0 as defined in conventional LPCC.

For the main experiment. combining CDCN with AP-derived BACC coefficients, we experimented with several *ad hoc* manual adjustments of CDCN parameters. (This was necessary because CDCN had been optimized to work with the outputs of conventional LPCC-based processing.) Finally an experimental technique called Histogram-based CDCN (H-CDCN)[53] was employed in place of the original CDCN, to improve the accuracy in discriminating the noise frames from their speech counterparts in training the CDCN normalization parameters.

Table 7-7 summarizes the performance of the mean-rate-derived and GSD-derived LPCC in conjunction with CDCN and H-CDCN processing. For each experiment run, we systematically varied the language weight of the recognizer and picked the best performance recorded in the CLSTK and CRPZM testing individually. (It is currently not possible to predict the optimal lan-

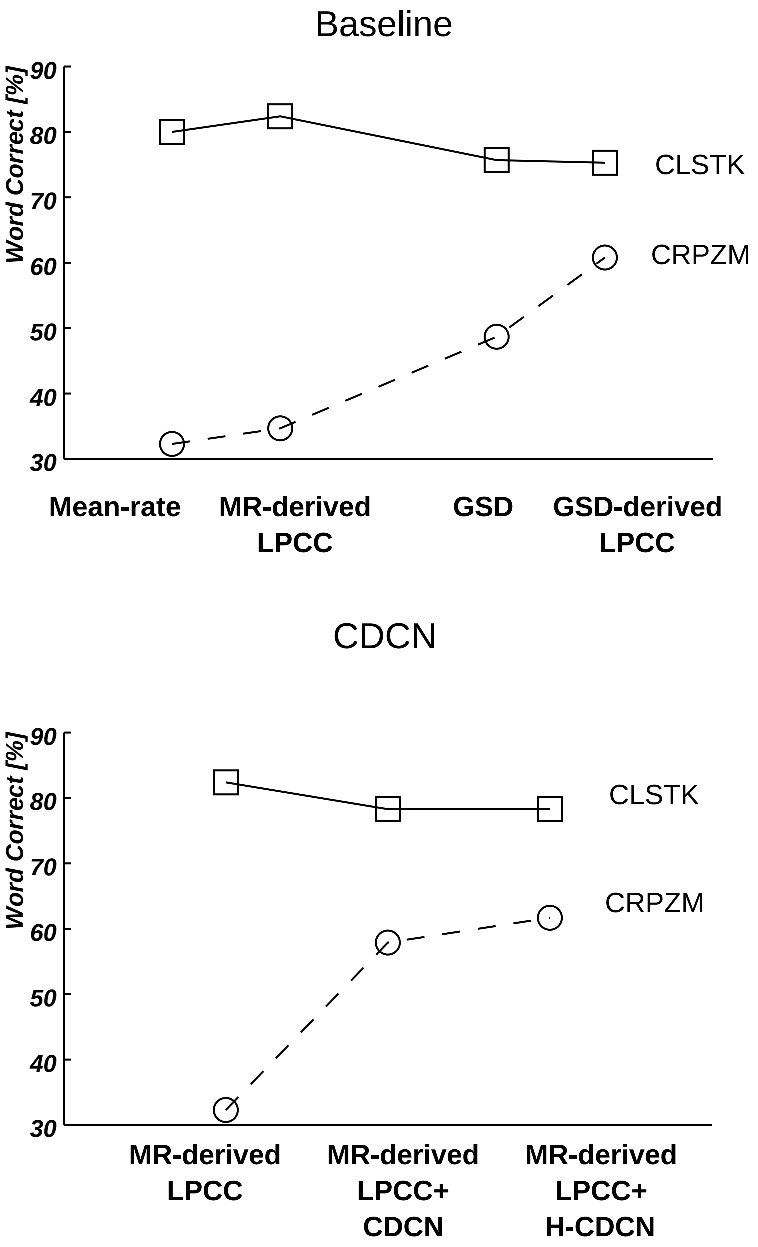

Figure 7-9 AN4 word recognition performance using the AP-derived LPCC front end with and without CDCN and Histogram-based CDCN. The system was trained with the CLSTK microphone and tested with the CLSTK and CRPZM microphones. Different set of parameters were considered for training the CDCN algorithm, and the most successful cases are presented here

guage weight for a given task and recognition system *a priori*, so exhaustive search for the best language weight is always necessary. However, as we already described, the relationship between raw word accuracy before counting insertion errors and the word accuracy after penalizing for insertions remained consistent regardless of the front-end, and was consistent with the trends we observed in evaluating the baseline system.

Processor Version	Language Weight	CLSTK Testing	CRPZM Testing
GSD	7.0	75.7%	48.7
GSD-derived LPCC	5.5	75.3	60.8
	6.5	75.8	60.8
Mean Rate(MR)	7.0	80.0	32.3
MR-derived LPCC	5.5	82.4	34.7
	9.5	79.3	39.5
MR-derived LPCC CDCN ver. 1	5.5	78.4	56.4
	7.0	78.3	57.9
MR-derived LPCC CDCN ver. 3	8.5	80.1	48.3
	9.0	79.4	48.4
MR-derived LPCC H-CDCN	7.5	79.0	60.7
	9.0	78.3	61.7

Table 7-7 AN4 word recognition performance using the AP-derived LPCC front end with and without CDCN and with Histogram-based CDCN. Different set of parameter values were considered for training the CDCN algorithm in each case, and the most successful cases are presented here.

We tried several different types of training procedures for the main experiment combining AP-derived LPCC and CDCN. We describe below were two of the most successful ones, in which CDCN improved the recognition accuracy significantly (to 57.9%) when the CRPZM microphone was used for testing. However, a few percent was lost when the CLSTK microphone was used for testing, maintaining the word recognition rate at 80.1%. In a previous study on the use the CDCN, it was reported that the CDCN caused a slight performance loss in the same microphone testing but the benefit gained from the CRPZM microphone outweighed by large[54]. The observation here was consistent to what was previously reported

In the last experiment, in which the AP-derived LPCC was normalized by the H-CDCN algorithm, we found that further improvement was possible in the CRPZM microphone case. The degradation from the unprocessed case in the CLSTK microphone testing became 2.4%, which would be a marginally significant difference.

7.2.3. AP-derived BACC Combined with CDCN

There is yet another way to compute cepstrum-like parameters derived from the downsampled output of Seneff model. The block diagram in Figure 7-10 describes the procedure to extract such parameters based on the pseudospectra.

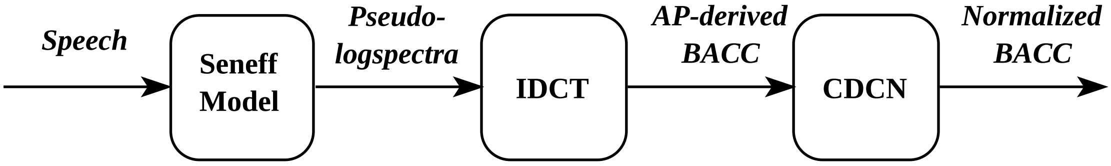

Figure 7-10 Block diagram of AP-derived BACC processing. Downsampled output from the Seneff model is converted using the inverse discrete cosine transform. The resulting feature vectors resemble BACC parameters because of the Bark-scale frequency spacing and the compressive nature of physiologically-motivated signal processing. CDCN post-processing is used to achieve further environmental compensation.

The output from the Seneff model can be characterized as a somewhat-compressed energy spectrum representation, and therefore can be directly transformed into the cepstrum-like representation of the speech by the inverse DCT. Let

$$S[k] \qquad ;k = 1, 2, \ldots, 40$$

denote the downsampled channel outputs of the Seneff model aligned in ascending order in frequency. As we reviewed in the previous chapter, the frequency spacing of the channels is uniform in the Bark scale, which results in a CF of 220 Hz at the low end and covers up to 6.3 kHz at the high end The inverse cosine transform of $S[k]$

$$c_i = \sum_{k=1}^{40} S[k] \cos\left[i\left(k - \frac{1}{2}\right)\frac{\pi}{40} \right] \qquad ;i = 1, 2, \ldots, p$$

results in a cepstrum-like series in the time domain. Due to the Bark-scaled center frequency spacing of the cochlea BPF in the Seneff model, the resulting pseudocepstra resemble BACC parameters. Nevertheless, it should be noted again that the individual channel signal, based on the pseudospectrum output, has already gone through various nonlinear signal processing operations in the NT stage of the auditory model, and it is expected to show robustness when compared against the original BACC. Now that we have transformed the feature vectors in the pseudocepstrum, further environmental normalization by an acoustical “pre-processing” algorithm such as CDCN is in order, based on the sample statistics of the parameters.

Table 7-8 shows the performance of AP-derived BACC parameters based on the mean-rate front-end and summarizes the results from several attempts to apply CDCN and H-CDCN. Here, again, the recognition accuracy using the CRPZM microphone was improved by about 5% using the BACC as compared with the baseline, while the performance with the CLSTK microphone was

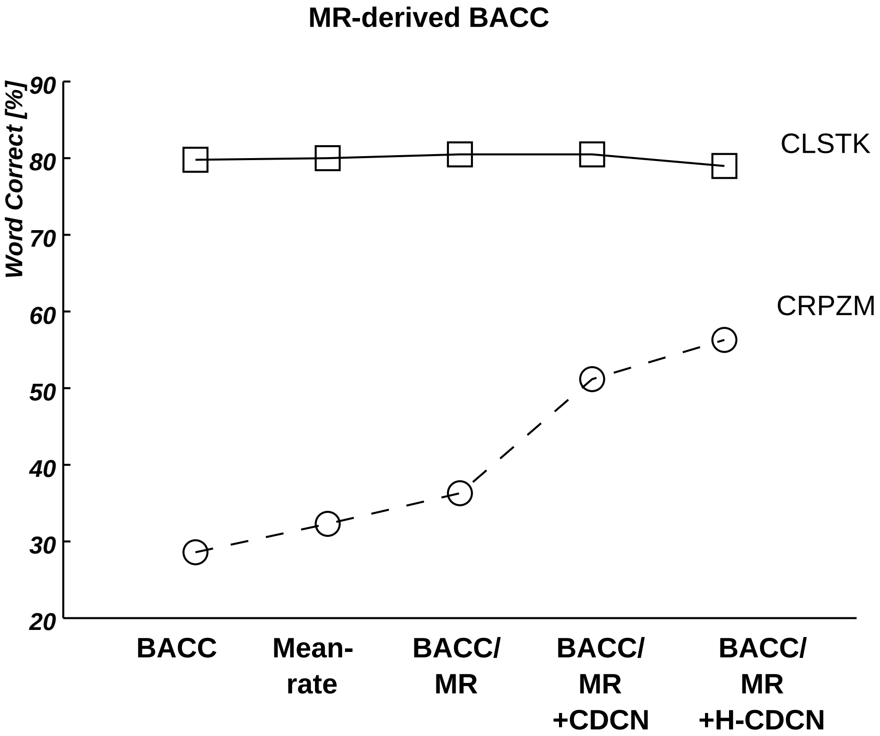

Figure 7-11 AN4 word recognition accuracy using the AP-derived BACC front-end with and without CDCN or Histogram-based CDCN. The system was trained with the CLSTK microphone and tested with the CLSTK and CRPZM microphones. Different set of parameter values were considered for training the CDCN algorithm, and the most successful cases are presented here. Also plotted for comparison are the baseline performance of the BACC front-end and the mean-rate front-end.

unaffected. This trend also is observed when the CDCN algorithm is applied. In the “best case”, testing with the CRPZM microphone, word recognition rate was as high as 60.7%. Although recorded in different CDCN runs, the CLSTK performance was consistently 79~80% for most cases. By using H-CDCN, we achieved promising levels of performance for both test conditions: 79.0% using the CLSTK microphone and 57.4% using the CRPZM microphone. Nevertheless, the CRPZM microphone performance did not quite match the best case among the CDCN runs with manual adjustment.

Discussion. For AP-derived BACC parameters, H-CDCN did not find the optimal cepstral normalization scheme that outdoes the best *ad hoc* attempt by manual adjustment of CDCN parameters. Although the H-CDCN is a fairly experimental extension to CDCN and is still undergoing refinements, we observe that the current version of H-CDCN seems to have a certain limitation in handling unusual histogram profiles of the energy term, c_0. Like the original implementation of CDCN, H-CDCN also assumes relatively benign bimodal histogram profile as the sample statistics of the underlying mixture Gaussian density representative of speech and noise. Therefore, estimates of the noise level become less appropriate as the part of c_0 histogram representing the noise deviates from a bell shape with a single distinct peak. As is shown in Figure 7-12, AP-derived cep-

Processor Version	Language Weight	CLSTK Testing	CRPZM Testing
Mean Rate(MR)	7.0	80.0%	32.3
MR-derived BACC	8.0	80.5	36.3
	9.0	79.1	37.4
MR-derived BACC CDCN ver. 1	7.5	76.5	48.8
MR-derived BACC CDCN ver. 3	6.0	77.3	56.7
	9.0	75.9	60.7
MR-derived BACC CDCN ver. 5	7.5	80.5	50.1
	8.5	80.5	51.2
MR-derived BACC CDCN ver. 6	7.0	80.5	41.9
	10.0	78.3	45.2
MR-derived BACC CDCN ver. 7	7.5	79.8	48.3
	8.0	79.2	48.5
MR-derived BACC H-CDCN	7.5	79.0	56.3
	9.5	77.6	57.4

Table 7-8 AN4 word recognition accuracy using the AP-derived BACC front-end with and without CDCN or Histogram-based CDCN.

stral parameters tend to produce c_0 histograms with two major spikes at low SNRs, which causes the identification of the SNR of noisy frames to become inaccurate. This aspect of the c_0 histogram can be attributed to the fact that the outputs of the various channels rest at a certain D.C. value to reflect the spontaneous firing rate, but they also tend to overshoot downwards and stay in the vicinity of zero while the channel input is rapidly dying out, thus causing two split peaks in the weak-intensity region of the histogram. One difficulty here is that there might be a good amount of c_0 information relevant to speech residing on both sides of the second peak of the histogram. Neither the current implementation of CDCN nor H-CDCN can handle this type of sample statistics adequately.

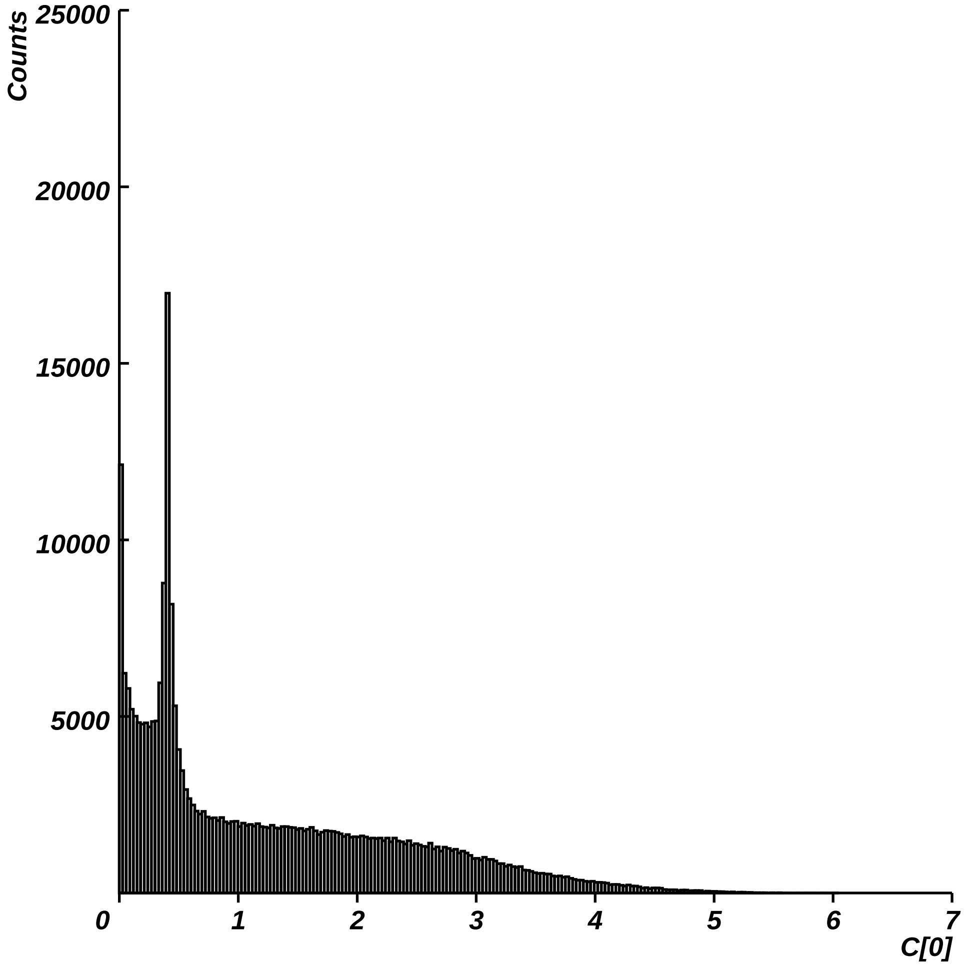

Figure 7-12 Histogram of c_0 coefficients of the AP-derived BACC for the AN4 training data.

To summarize, whether we manually try different training parameters to find the optimal CDCN scheme or whether we resort to H-CDCN, the underlying sample statistics of the c_0 histograms are most likely the major obstacle that prevent us from using CDCN in an optimal fashion in conjunction with the AP-derived cepstral parameters. Nevertheless, we have shown that recognition accuracy using the CRPZM microphone and AP-derived cepstra is further improved by applying CDCN or H-CDCN.

7.3 Summary

In this chapter we described procedures with which we can combine acoustical preprocessing algorithms and physiologically-motivated signal processing both in the waveform and parameter domains.

In the waveform domain, we tried two different procedures, waveform resynthesis from CDCN outputs and waveform reconstruction from the mean-rate and GSD outputs of the Seneff model. CDCN successfully normalized speech from the CRPZM microphone and therefore improved the performance of the Seneff model. However, it did not exceed the best performance previously achieved by use of the LPCC front-end in conjunction with CDCN. In waveform reconstruction, the nonlinear stages of the Seneff did not appear to produce further improvements in recognition accuracy. However, we gained much useful insight about these techniques.

In the parameter domain, we first applied SDCN to the mean-rate outputs, and later applied CDCN to the cepstral coefficients of Seneff model parameters derived either by all-pole modelling or the inverse DCT. Both mean-rate and GSD outputs in their raw feature form, were known to perform as well as LPCC with CDCN and better than LPCC with SDCN in additive noise conditions. This was further improved by applying the SDCN algorithm to the mean-rate front-end in the spectral domain.

The derived cepstral coefficients were more robust than those in the raw feature representation. Performance in noisy testing conditions were further improved by using CDCN (or H-CDCN), while performance in clean speech was maintained relative to the one without CDCN. H-CDCN was found to be more useful than CDCN in the all-pole modelling of the mean-rate output. Also the short-term energy estimated from the Seneff model spectra was found to be more robust than the conventional energy measurement.

Finally, the CDCN training procedure will need to undergo further modification for optimal performance when used in conjunction with AP-derived cepstra. The sample statistics of the energy term are rather distinctly from the bimodal Gaussian mixture density which the training procedure assumes to be valid.

Chapter 8 Summary

8.1. Major Contributions

We investigated the use of physiologically-motivated signal processing to provide environmental robustness in speech recognition using real and simulated forms of acoustical degradation. Our major contributions are as follows:

1. We evaluated the robustness of Seneff model as a representative example of physiologically-motivated signal processing. We found that both the mean-rate and GSD outputs of the Seneff model provided improvements in recognition accuracy compared to the conventional cepstral representation of speech both when the speech was subjected to artificially-added noise and when the speech was modified by unknown linear filtering. We found that the mean-rate front-end generally performed well for all conditions considered, while the GSD outputs were particularly effective for low SNRs in conjunction with spectral tilt. We investigated the number of principal components needed for the Seneff model to provide robustness in real degradations and found that 5 components for the mean-rate outputs and 10 components for the GSD outputs were sufficient.
2. We studied in a systematic and parametric manner the extent to which individual components of the NT stage of the Seneff model contribute to environmental robustness. We evaluated the significance of each module in terms of its impact on recognition accuracy in the presence of real and simulated degradation. We found that the most important NT component after the rectifier was short-term adaptation. We found that other components are also important inasmuch as we could not eliminate any of them without sacrificing recognition accuracy in the other environmental conditions considered.
3. We showed that the recognition accuracy provided by physiologically-motivated signal processing can (in some circumstances) be further improved by combination with acoustical preprocessing. These two approaches to robust recognition can be combined in two ways: by normalizing the derived cepstra or by normalizing the input speech waveform.
4. We successfully developed a method to resynthesize speech from its cepstral representation, and we demonstrated the usefulness of this procedure for speech recognition. We compensated for occasional instabilities in the synthesis filter by reflecting poles into the inside of the unit circle without loss of recognition accuracy.
5. We also developed a method for reconstructing speech waveforms from the outputs of the Seneff model. Although this procedure did not improve recognition accuracy, it does produce intelligible speech, and the procedure may provide a useful amount of speech enhancement.

8.2. Suggestions for Future Work

The results of our research are leading us to look into several areas for further investigation for tomorrow:

1. Now that it is possible for us to hear the effect of cepstral normalization, it would be worthwhile to determine the extent to which our waveform resynthesis can provide a useful amount of enhancement for speech perceived by humans. So far most studies indicate that classical single-channel speech enhancement techniques may produce speech that is subjectively more appealing but that is not more intelligible using any of several standard figures of merit[55]. It is possible that the resynthesis of degraded speech either from cepstral coefficients modified by methods like CDCN or from the outputs of the nonlinear NT stages of the Seneff model may be more intelligible than unprocessed speech.

2. In combining CDCN and physiologically-motivated processing in the parameter domain, we encountered a problem in optimizing the training of the CDCN parameters. Specifically, the probability density function of energy coefficients of the outputs of the auditory model is fundamentally different from the Gaussian density function describing the energy coefficients produced by conventional LPCC analysis. This difference impairs the effectiveness of algorithms like H-CDCN. We believe that the CDCN algorithm could be much more effective if it could be modified to process power coefficients with non-Gaussian distributions in an appropriate fashion.

3. We believe it would be very useful if we conducted a comparative study of GSD, Correlogram, TQTI, as they are similar in the attempt to reinforce the periodicity of channel waveforms. Analysis of their strength and weakness has not been done yet, and it would enable us to gain more insight about the importance of synchrony detection.

4. The comparisons of PLP analysis and the use of principal components of the Seneff model imply that only a small number of parameters are needed for a speaker-independent task such as AN4. It is also possible that this data trend is an idiosyncracy of the PLP and Seneff models. The use of principal component analysis for more commonly-used feature sets such as MFCC would bring us more confidence in making judgements about the generality of their value. Also the use of linear discriminant analysis instead of principal component analysis would be promising, as it transforms the feature parameter space to give maximum separability.

5. One of the most difficult aspects of this research has been the lack of an obvious way of evaluating the individual merit of front ends short of measuring their actual contribution to recognition accuracy. In a highly integrated system like SPHINX, there are many system parameters to be considered that can affect an experimental outcome in addition to the specific aspect of front-end processing that manipulated in a given experiment.

Appendix A Derivation of LPCC and Its Inverse Formula

This paragraph briefly describes the procedures to compute the LPC cepstrum parameters from the LPC parameters and its inverse. The derivation of the LPCC is formulated in the z-domain by computing the impulse response of the complex logarithm of the underlying LPC system, which is analogous to the computation of cepstrum in the discrete Fourier transform domain.

The procedure to compute LPCC parameters is as follows. We start with the transfer function of the p-th order LPC system:

$$H(z) = \sum_{n=0}^{+\infty} h[n] z^{-n} = \frac{G}{A(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}}$$

Take the derivative of the complex polynomial $ln(H(z))$ with respect to $\rho = z^{-1}$ to obtain:

$$\frac{\partial}{\partial\rho} ln(H(\rho)) = \frac{\partial}{\partial\rho}[ln(G) - ln(A(\rho))] = \frac{\sum_{l=1}^{p} l a_l \rho^{l-1}}{1 - \sum_{i=1}^{p} a_i \rho^i}$$

If $H(z)$ has all of its poles inside the unit circle, then $ln(H(z))$ is a unilateral z-polynomial

$$C(z) = \sum_{i=0}^{+\infty} c_i z^{-i}$$

By taking the derivative of $C(z)$ with respect to ρ and equating it to the above result:

$$\sum_{j=1}^{+\infty} j c_j \rho^{j-1} = \frac{\sum_{l=1}^{p} l a_l \rho^{l-1}}{1 - \sum_{i=1}^{p} a_i \rho^i}$$

or

$$\left(\sum_{j=1}^{+\infty} j c_j \rho^{j-1}\right)\left(1 - \sum_{i=1}^{p} a_i \rho^i\right) = \sum_{l=1}^{p} l a_l \rho^{l-1}$$

By comparing the coefficients of the power series of ρ on both sides, we obtain the following recursive formulae for c_i. Also c_0 is determined by the constant term of the original definition of $H(z)$

$$c_i = \begin{cases} ln(G) & ;i=0 \\ a_1 & ;i=1 \\ a_i + \sum_{j=1}^{i-1} \frac{i-j}{i} c_{i-j} a_j & ;1 < i \le p \\ \sum_{j=1}^{p} \frac{i-j}{i} c_{i-j} a_j & ;i > p \end{cases}$$

The inverse procedure is easily found as follows:

$$a_i = \begin{cases} c_1 & ;i=1 \\ c_i - \sum_{j=1}^{i-1} \frac{j}{i} a_{i-j} c_j & ;1 < i \le p \end{cases}$$

Appendix B Frequency Warping by Bilinear Transform

The frequency warping is a technique motivated by the need of non-uniform frequency sampling and is realized by using the bilinear transform

$$Z^{-1} = \frac{z^{-1} - a}{1 - az^{-1}} \qquad ;|a| < 1$$

which defines the relationship between the original and transformed frequency axes ω and Ω for

$$z = e^{j\omega}$$
$$Z = e^{j\Omega}$$

to satisfy the following:

$$\Omega = \omega + 2\tan^{-1}\left(\frac{a\sin\omega}{1 - a\cos w}\right)$$

It was shown by Oppenheim and Johnson that its digital implementation takes the form of the all-pass network of filters[56]. The sequence $\{c'_n\}$, of which spectrum is warped in frequency from

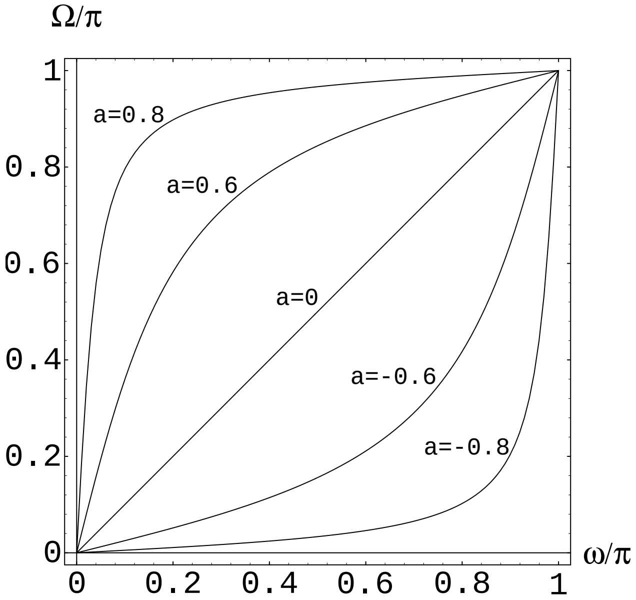

Figure B-1 Frequency warping by bilinear transform. Labeled axes are normalized DTFT frequencies for the original and the transformed. Positive values for the warping parameter a result in the compression in the high frequency region of the original, and the negative a values have the opposite effect.

the original $\{c_n\}$, is obtained by sampling the output of the network at time $n = 0$ to the time-reversed sequence of $\{c_n\}$, as is shown in Figure B-1

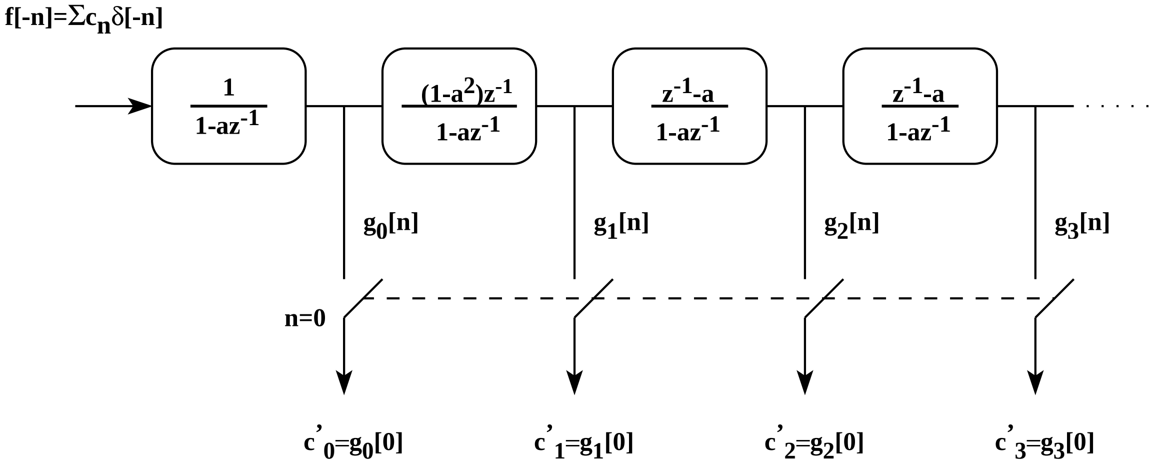

Figure B-1 Discrete time implementation of frequency warping by bilinear transform.

Also it should be noted that the synthesis of cascaded frequency warping operations is of the following form, i.e. with

$$W^{-1} = \frac{Z^{-1} - b}{1 - bZ^{-1}}$$

$$Z^{-1} = \frac{z^{-1} - a}{1 - az^{-1}}$$

we realize that

$$W^{-1} = \frac{Z^{-1} - b}{1 - bZ^{-1}} = \frac{\frac{z^{-1} - a}{1 - az^{-1}} - b}{1 - b\frac{z^{-1} - a}{1 - az^{-1}}} = \frac{z^{-1} - \left(\frac{a + b}{1 + ab}\right)}{1 - \left(\frac{a + b}{1 + ab}\right) z^{-1}}$$

In the LPCC front-end of the SPHINX speech recognition system, the bilinear transform with the warping parameter $a = 0.6$ was used. From the synthesis formula derived above, it is obvious that another application of the bilinear transform with the parameter $a = -0.6$ could be used to undo the effect.

Bibliography

[1] K.-F. Lee, “Large-Vocabulary Speaker-Independent Coutinuous Speech Recognition: The SPHINX System”, PhD dissertation, Carnegie Mellon University, April 1988.

[2] A. Acero, “Acoustical and Environmental Robustness in Automatic Speech Recognition”, PhD dissertation, Carnegie Mellon University, September 1990.

[3] M. J. Hunt and C. Lefèbvre, “Speech Recognition Using an Auditory Model with Pitch-Synchronous Analysis”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 1987, pp. 813--816.

[4] M. J. Hunt and C. Lefèbvre, “Speaker Dependent and Independent Speech Recognition Experiments with an Auditory Model”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 1988, pp. 215--218.

[5] O. Ghitza, “Robustness against Noise: The Role of Timing-Synchrony Measurement”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 1987, pp. 2372--2375.

[6] H. M. Meng and V. W. Zue, “A Comparative Study of Acoustic Representations of Speech for Vowel Classification Using Multi-Layer Perceptrons”, *International Conference on Spoken Language Processing*, The Acoustical Society of Japan, 1990, pp. 1053--1056.

[7] S. Seneff, “A joint synchrony/mean-rate model of auditory speech processing”, *Journal of Phonetics*, Vol. 16, January 1988, pp. 55--76.

[8] K.-F. Lee, *Automatic Speech Recognition--The Development of the SPHINX System*, Kluwer Academic Publishers, 1989.

[9] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, Springer-Verlag, 1976.

[10] B. S. Atal, “Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification”, *Journal of the Acousticstical Society of America*, Vol. 55, No. 6, 1974, pp. 1304--132.

[11] Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design”, *IEEE Transactions on Communication*, Vol. COM-28, No. 1, January 1980, pp. 84--95.

[12] F. Jelinek, “Continuous Speech Recognition by Statistical Methods”, *Proceedings of the IEEE*, Vol. 64, No. 4, April 1976, pp. 532--556.

[13] B. Juang, “Speech Recognition in Adverse Environments”, *Computer Speech and Language*, Vol. 5, 1991, pp. 275--294.

[14] D. Mansour and B. Juang, “A Family of Distortion Measures Based upon Projection Operation for Robust Speech Recognition”, *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. 37, November 1989, pp. 1659--1671.

[15] S. Lerner and B. Mazor, “Telephone Channel Normalization for Automatic Speech Recognition”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol. 1, March 1992, pp. I.261--I.264.

[16] B. Widrow, et. al., “Adaptive Noise Cancelling: Principles and Applications”, *Proceedings of the IEEE,* Vol. 63, 1975, pp. 1692--1716.

[17] Y. Ephraim, et. al, “A Linear Predictive Front-End Processor for Speech Recognition in Noisy Environments”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* April 1987, pp. 1324--1327.

[18] R. F. Lyon, “A Computational Model of Filtering, Detection, and Compression in the Cochlea”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, 1982, pp. 1282--1285.

[19] S. Seneff, “Pitch and Spectral Analysis of Speech Based on an Auditory Synchrony Model”, PhD dissertation, Massachusetts Institute of Technology, January 1985.

[20] N. Y.-S. Kiang, T. Watanabe, E. C. Thomas, and L. F. Clark, *Discharge Patterns of Single Fibers in the Cat's Auditory Nerve,* MIT Press, Cambridge, Mass., 1965, Research Monograph 35.

[21] E. Zwicker, “Subdivision of the audible frequency range into critical bands (frequenzgruppen)”, *Journal of the Acousticstical Society of America,* Vol. 33, No. 2, February 1961, pp. 248.

[22] R. S. Goldhor, “Representation of Consonants in the Peripheral Auditory System: A Modeling Study of the Correspondence between Response Properties and Phonetic Features”, PhD dissertation, Massachusetts Institute of Technology, February 1985.

[23] C. G. M. Fant, “Acoustic Description and classification of phonetic units”, *Ericsson Technics,* No. 1, 1959.

[24] O. Ghitza, “Speech Analysis/Synthesis Based on Matching the Synthesized and the Original Representations in the Auditory Nerve Model”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, 1986, pp. 1995--1998.

[25] S. Shamma, “The acoustic features of speech sounds in a model of auditory processing: vowels and voiceless fricatives”, *Journal of Phonetics,* Vol. 16, January 1988, pp. 77--91.

[26] M. Slaney, “Lyon's Cochlea Model”, Apple Technical Report 13, Apple Computer, Inc., November 1988.

[27] E. D. Young and M. B. Sachs, “Representation of steady-state vowels in the temporal aspects of the discharge patters of populations of auditory-nerve fibers”, *Journal of the Acousticstical Society of America,* Vol. 66, 1979, pp. 1381--1403.

[28] O. Ghitza, “Auditory Nerve Representation as a Front-End for Speech Recognition in a Noisy Environment”, *Computer Speech and Language,* Vol. 2, No. 1, 1987, pp. 109--130.

[29] O. Ghitza, “Temporal Non-Place Information in the Auditory-Nerve Firing Patterns as a Front-End for Speech Recognition in a Noisy Environment”, *Journal of Phonetics,* Vol. 16, January 1988, pp. 109--123.

[30] M. J. Ross et. al., “Average magnitude difference function pitch extracter”, *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. ASSP-22, No. 5, October 1974, pp. 353--362.

[31] J. A. Moorer, “The Optimum Comb Method of Pitch Period Analysis of Continuous Digitized Speech”, *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. ASSP-22, No. 5, October 1974, pp. 330--338.

[32] R. F. Lyon, “A Computational Model of Binaural Localization and Separation”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, 1983, pp. 1148--1151.

[33] M. Slaney and R. F. Lyon, “A Perceptual Pitch Detector”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, 1990, pp. 357--360.

[34] J. Holdsworth, J. L. Schwartz, F. Berthommier and R. D. Patterson, “A multi-representation model for auditory processing of sounds”, *Auditory Physiology and Perception--9th International Symposium on Hearing,* 1991.

[35] M. P. Vea, “Multisensor Signal Enhancement for Speech Recognition”, Master's thesis, Carnegie Mellon University, September 1987.

[36] J. Picone, G. R. Doddington, D. S. Pallett, “Phone-mediated word alignment for speech recognition evaluation”, *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. ASSP-38, No. 3, March 1990, pp. 559--562.

[37] D. S. Pallett, W. M. Fisher and J. G. Fiscus, “Tools for the analysis of benchmark speech recognition tests”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, April 1990, pp. 97--100.

[38] L. Gillick and S. J. Cox, “Some statistical issues in the comparison of speech recognition algorithms”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* IEEE, May 1989, pp. 532--535.

[39] J. N. Marcus, “Significance tests for comparing speech recognizer performance using small test sets”, *EUROSPEECH 89,* September 1989, pp. 465--468.

[40] J. G. Fiscus, “The NIST Benchmark Speech Recognition System Scoring Program, Version 3.0”.

[41] S. B. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 4, August 1980, pp. 357--366.

[42] B. Chigier, “The effects of the telephone network on phoneme classification”, *Proceedings 1991 IEEE Workshop on Automatic Speech Recognition*, 1991.

[43] H. Hermansky, “Perceptual linear predictive (PLP) analysis for speech”, *Journal of the Acousticstical Society of America*, Vol. 87, No. 4, April 1990, pp. 1738--1752.

[44] H. Hermansky and J. C. Junqua, “Optimization of perceptually-based ASR front-end (automatic speech recognition)”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, April 1988, pp. 219--222.

[45] M.-Y. Hwang, “Unpublished”.

[46] C. R. Jankowski, Jr., “A Comparison of Auditory Models for Automatic Speech Recognition”, Master's thesis, Massachusetts Institute of Technology, May 1992.

[47] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.

[48] T. M. Sullivan and R. M. Stern, “Multi-Microphone Correlation-Based Processing for Robust Speech Recognition”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 1993, pp. II.91--II.94.

[49] T. M. Sullivan, “Unpublished”.

[50] R. M. Stern, F.-H. Liu, Y. Ohshima, T. Sullivan, and A. Acero, “Multiple Approaches to Robust Speech Recognition”, *Proceedings of Speech and Natural Language Workshop*, Defense Advanced Research Projects Agency, February 1992, pp. 274--279.

[51] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, 1992.

[52] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[53] N. Hanai, “Speech Recognition in the Automobile”, Master's thesis, Carnegie Mellon University, May 1993.

[54] F.-H. Liu, R. M. Stern, X. Huang, and A. Acero, “Efficient Cepstral Normalization for Robust Speech Recognition”, *DARPA Human Language Technology Workshop*, March 1993.

[55] J. S. Lim, editor, *Speech Enhancement*, Prentice-Hall, 1983.

[56] A. Oppenheim, D. Johnson, and K. Steiglitz, “Computation of Spectra with Unequal Resolution Using the Fast Fourier Transform”, *Proceedings of the IEEE(Lett.),* Vol. 59, February 1971, pp. 299--301.