# Acoustical and Environmental Robustness in Automatic Speech Recognition

**Alejandro Acero**

September 13, 1990

Department of Electrical
and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

A Dissertation submitted to the Graduate School in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.



This research was partly sponsored by a National Science Foundation Graduate Fellowship, and by Defense Advanced Research Projects Agency Contract N00039-85-C-0163. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, or the US Government.

# Abstract

This dissertation describes a number of algorithms developed to increase the robustness of automatic speech recognition systems with respect to changes in the environment. These algorithms attempt to improve the recognition accuracy of speech recognition systems when they are trained and tested in different acoustical environments, and when a desk-top microphone (rather than a close-talking microphone) is used for speech input. Without such processing, mismatches between training and testing conditions produce an unacceptable degradation in recognition accuracy.

Two kinds of environmental variability are introduced by the use of desk-top microphones and different training and testing conditions: additive noise and spectral tilt introduced by linear filtering. An important attribute of the novel compensation algorithms described in this thesis is that they provide *joint* rather than independent compensation for these two types of degradation.

Acoustical compensation is applied in our algorithms as an additive correction in the cepstral domain. This allows a higher degree of integration within SPHINX, the Carnegie Mellon speech recognition system, that uses the cepstrum as its feature vector. Therefore, these algorithms can be implemented very efficiently. Processing in many of these algorithms is based on instantaneous signal-to-noise ratio (SNR), as the appropriate compensation represents a form of noise suppression at low SNRs and spectral equalization at high SNRs.

The compensation vectors for additive noise and spectral transformations are estimated by minimizing the differences between speech feature vectors obtained from a "standard" training corpus of speech and feature vectors that represent the current acoustical environment. In our work this is accomplished by a minimizing the distortion of vector-quantized cepstra that are produced by the feature extraction module in SPHINX.

In this dissertation we describe several algorithms including the *SNR-Dependent Cepstral Normalization*, (SDCN) and the *Codeword-Dependent Cepstral Normalization* (CDCN). With CDCN, the accuracy of SPHINX when trained on speech recorded with a close-talking microphone and tested on speech recorded with a desk-top microphone is essentially the same obtained when the system is trained and tested on speech from the desk-top microphone.

An algorithm for frequency normalization has also been proposed in which the parameter of the bilinear transformation that is used by the signal-processing stage to produce frequency warping is adjusted for each new speaker and acoustical environment. The optimum value of this parameter is again chosen to minimize the vector-quantization distortion between the standard environment and the current one. In preliminary studies, use of this frequency normalization produced a moderate additional decrease in the observed error rate.

# Acknowledgments

First, I would like to thank Richard Stern, my thesis advisor, for his invaluable support and guidance. His thorough writing abilities have helped me make this dissertation a great deal better. Over the years we have developed a relationship that has contributed to making my stay at CMU a wonderful experience.

I am also indebted to the members of my committee: Raj Reddy, Kai-Fu Lee, Vijaya Kumar and Jose Moura for their help. Raj has followed closely my research and he has suggested many ideas. Kai-Fu has helped me understand some of the mysteries in the SPHINX system. All of them gave me some useful suggestions to improve this document.

I am very grateful to the members of the speech group. Shuji Morii's studies were the motivation for this work. Bob Weide patiently collected the databases I used. My officemates Tom Sullivan and Yoshiaki Ohshima had to stand my bad jokes for two years; I also thank them for making the office an enjoyable place to be.

Finally I would like to thank my family and friends for their support. My brother was always there to give me good advice. My parents were always with me when I needed them and they have played a major role in these thesis years. They made big problems look insignificant and they definitely share some of the success of this thesis. I will always be grateful to them.

# 1 Introduction

Unconstrained automatic speech recognition (ASR) is a very difficult problem. Early ASR systems obtained a reasonable performance by artificially constraining the problem. Those systems were speaker dependent, and dealt with isolated speech for small vocabularies, while current speech recognizers produce higher performance under less constrained environments. SPHINX (Lee *et al.* [45]), the system recently developed at CMU, is the first speaker-independent large-vocabulary continuous-speech recognition system. The technology has now reached the point where ASR systems may become useful in everyday applications although there are other considerations that must be addressed for speech to become a viable and facile man-machine communication medium:

- **Spontaneous speech**. For a system to be useful in a day-to-day application it has to accept spontaneous speech. Most current ASR systems deal with read speech only because spontaneous speech is often ungrammatical and ill-structured.
- **Semantics and pragmatics in speech understanding**. For a specific task, the perplexity, average number of words active at any given time, can be reduced substantially if semantic and pragmatic information is used in addition to a grammar.
- **Acoustical and environmental robustness**: ASR systems exhibit unacceptable degradations in performance when the acoustical environments used for training and testing the system are not the same. It would be desirable to have a system that works independently of recording conditions (using different rooms, microphones, noise levels).

In this dissertation we focus on the last issue, the robustness with respect to changes in the acoustical environment, and we describe several algorithms that make SPHINX more *environment independent*.

## 1.1. Acoustical Environmental Variability and its Consequences

In this section we discuss the different causes of acoustical and environmental variability. There are some attributes of the environment that remain relatively constant through the course of an utterance such as the recording equipment, the amount of room reverberation, and the acoustical characteristics of the particular speaker using the system. Other factors, like the noise and signal levels, will be assumed to vary slowly compared to the rate at which speech changes.

Current HMM[1] recognizers usually focus on short-term attributes of the speech signal (over segments of about 20 ms that are called frames) while long-term properties are often ignored. We will discuss several different external factors that change the environment and therefore affect the performance of a speech recognition system: input level, additive noise, spectral tilt, physiological differences, and interference by the speech of other speakers (the cocktail party effect).

### 1.1.1. Input Level

Input level changes from utterance to utterance and even within the same utterance. Speakers normally do not speak with the same volume, distance from, and orientation to the microphone. Gain normalization cannot be done on a frame-by-frame basis since the input level is a long-term feature of speech.

### 1.1.2. Additive Noise

The performance of speech recognition systems degrades drastically when training and testing are carried out with different noise levels. When the signal-to-noise ratio (SNR)[2] is less than +10 dB the speech is severely corrupted and not even training and testing on the same noisy environment can ameliorate the problem.

In most cases background noise can be modeled as an additive stationary perturbation that is uncorrelated with the signal. We are excluding from this analysis non-stationary perturbations frequent in an office environment such as door slams, key strokes and other conversations. The stationarity of the background noise is another long-term property of our corrupted speech.

---

[1]HMM, Hidden Markov Models, is the dominant technology in automatic speech recognition. See Chapter 2 for an overview.

[2]Signal-to-noise Ratio is the ratio between the power of the signal and the power of the noise, and it is usually given in dB.

### 1.1.3. Spectral Tilt

Spectral tilt is the distortion observed in the spectrum when the speech signal is passed through an LTI (linear time-invariant) filter. In the log-spectral domain the net effect of spectral tilt is an additive offset to the transfer function. There are various sources of spectral tilt:

- **Speaker-specific spectral characteristics**. Differences in vocal tract physiology will lead to different long-term average frequency responses. Accurate speaker verification and identification systems have been built that use long-term spectral averages as their features.
- **Speech Styles**. It has been discovered recently [7] that spectral tilt is one of the major effects of the use of different speech styles (*e.g.* normal, soft, fast, and shouted speech, and speech spoken in the presence of noise [Lombard-effect speech]).
- **Room acoustics and reverberation**. The effect of reverberation can be modeled as a linear filter that depends on the geometry and materials of the room and the speaker location.
- **Recording equipment**. If a different microphone is used, the overall transfer function will change.

Spectral tilt is a major cause of performance degradation. The spectral distortion measures[3] used in speech recognition are severely affected by spectral tilt.

### 1.1.4. Physiological Differences

Differences in the physiology of the vocal tract will produce variability in the speech signal. Although formant frequencies are probably among the most invariant features in vowel discrimination, there is great variability among speakers due to differences in vocal tract sizes and shapes.

It is well known that the formant frequencies for a voiced sound will depend on the phonetic context (*short-term variability*) as well as on the anatomical characteristics of the speaker (*long-term variability*). Male voices exhibit lower formants than those of females, who in turn exhibit lower formants than those of children. The nominal resonant frequencies of the vocal tract depend strongly on the size of the vocal tract as well as on some other anatomical parameters.

---

[3]Spectral distortion measures typically compute $\int_{-\pi}^{\pi} \{f[S_a(\omega)] - f[S_b(\omega)]\}^2 d\omega$ where the function $f(x)$ is $x$ or a compressing function of $x$ (the logarithm is a popular choice).

### 1.1.5. Interference from Other Speakers

The interference that takes place when different speakers talk simultaneously is sometimes referred to as the cocktail party effect. In most speech recognition systems the input signal is modeled as an excitation (specifically an impulse train or white noise) driving a time-varying all-pole filter. Since this model assumes that only a single voice is present, interference from other speakers cause a dramatic degradation in recognition accuracy. While automatic recognition systems fail, humans do remarkably well in attending to and interpreting the speech of the desired speaker.

## 1.2. Previous Research in Signal Processing for Robust Speech Recognition

Robust speech recognition is a young and rapidly growing field. Most of the early work toward robustness has been derived from classical techniques developed in the context of speech enhancement (Lim [49] offers a good summary of those techniques). In this section we describe different kinds of techniques used to enhance speech in the presence of additive noise.

### 1.2.1. Techniques Based on an Autoregressive Model

Autoregressive analysis (AR), is a set of techniques that assume that the signal spectrum can be represented by the all-pole transfer function

$$A(z) = \frac{G}{1 + \sum_{i=1}^{p} \alpha_i z^{-i}} \tag{1.1}$$

When such a signal is corrupted by additive noise, zeroes will appear as well as poles in the transfer function representing the signal. The goal of these approaches is to find the autoregressive (AR) parameters of the speech signal and the noise. Lim [48] proposed a speech enhancement technique that iterated back and forth between an estimate of the clean waveform obtained by Wiener filtering of the noisy speech and the AR parameters obtained from that clean waveform. Proof of the convergence of these techniques was later provided by Feder *et al.* [21]. Although this processing produced an increase in SNR, the processed speech was less intelligible, and the resulting estimates exhibited a large variance.

Hansen and Clements [32] [33] extended Lim's work by applying *inter-frame* and *intra-frame* constraints to reduce the variance of the estimates of the restored speech. When applied to speech recognition, they obtained a somewhat higher recognition rate. The work of Mansour and Juang [52] is also based on an AR model of speech.

The main problem with these approaches is that small inaccuracies of the all-pole model are magnified by this algorithm at high noise levels. Also, the restored speech may exhibit artificial resonances that do not correspond to real speech frames.

### 1.2.2. Techniques Based on Manipulation of Distortion Measures

A number of researchers have attempted to combat additive noise and spectral tilt by developing special distortion measures that are more robust to this type of variability. Gray *et al.* [30] presents an early study on different distortion measures for speech processing which favors the cepstral distance[4]. A more recent study was presented by Nocerino *et al.* [59].

Instead of using the Euclidean distance[5], some authors have proposed Mahalanobis distance[6] distances for cepstral feature vectors. Tokhura [78] showed that since different cepstral coefficients are uncorrelated with each other, it suffices to use a weighted Euclidean distance. The weights were the inverse of the variance of each cepstral coefficient obtained from the pooled data. For recordings on dial-up telephone lines this method proved to be quite effective.

Juang *et al.* [41] proposed the use of bandpass *liftering*[7], in which the cepstral coefficients were weighted by a raised sine function. They argued that reducing the contribution of low-order cepstral coefficients was beneficial because of their large variance, and that reducing the contribution of the highest-order cepstral coefficients improved the discrimination capabilities of ASR systems. Again this algorithm proved effective for dial-up telephone lines for digit recognition.

Other types of lifters were proposed by Hermansky [34], Itakura and Umezaki [39]. Junqua and Wakita [42] did a comparative study of lifters for robust recognition.

Mansour and Juang [52] proposed a set of distortion measures that gives consideration to the angle between two vectors, rather than just the norm of the difference. This distortion measure was very effective for combating dial-up telephone speech corrupted with additive synthetic noise for an alphanumeric isolated-word task.

---

[4]The *cepstrum*, originally proposed by Bogert *et al.* [4], is the inverse Fourier transform of the logarithm of the spectrum. The cepstrum is a popular choice for speech recognition front-ends.

[5]The *Euclidean distance* between two vectors $\mathbf{x}$ and $\mathbf{y}$ is $d = [\sum (\mathbf{x}[k] - \mathbf{y}[k])^2]^{1/2}$.

[6]The *Mahalanobis distance* between vectors $\mathbf{x}$ and $\mathbf{y}$ is $d = (\mathbf{x} - \mathbf{y})^T \mathbf{C} (\mathbf{x} - \mathbf{y})$ where $\mathbf{C}$ is the inverse of the covariance matrix of $\mathbf{x}$ and $\mathbf{y}$.

[7]The word *liftering* comes from filtering with the first syllable reversed. The term was coined by Bogert *et al.* [4] in his work defining the cepstrum (spectrum backwards) as the inverse Fourier transform of the logarithm of the Fourier transform.

Soong and Sondhi [73] proposed a frequency-weighted measure that accounted for the broadening of the peaks at low SNR. Goncharoff and Chandran [29] and Noda [60] did similar studies based on spectral warping.

Although relatively successful, one paradox of all these weighted Euclidean distortions is that they intend to combat distortions due to a shift in the mean (spectral tilt) by a variance normalization. Although these authors reported improvement on speech recorded over many different telephone lines over the case of no processing, we found in pilot experiments that Juang's liftering, Tokhura's weighted distance and Hermansky RPS distortion did not produce any useful improvement in recognition accuracy for our system using our database, perhaps because the frequency warping transformation used in SPHINX alters the variance of the cepstral coefficients.

### 1.2.3. The Use of Auditory Models

Since the human auditory system is very robust to changes in acoustical environments, some researchers have attempted to develop signal-processing schemes that mimic the functional organization of the peripheral auditory system. For example, Seneff [70] and Zue *et al.* [83], Hunt and Lefebvre [37], Ghitza [25] [26] and Cohen [9] all used models based on the human auditory system as front-end processors for their recognition systems.

While these methods appear to be able to produce some improvement in robustness for speech in noise, the mechanisms that enable this to occur are not well understood. In fact, Ghitza [26] found that the use of DFT[8] coefficients resulted in higher accuracy and robustness than the highly complex filter-bank. He found the power of the model for robustness to be in the non-linear stages. In later work, Hunt and Lefebvre [38] analyzed the performance of different weighting schemes that used linear discriminant analysis (LDA) to combine heterogeneous speech features. In their experiments, a front-end based on LDA techniques attained a higher degree of robustness than the auditory model they described in [37] for the same task while being substantially more computationally efficient.

Cohen [9] designed a front-end based on some of the functions of the peripheral auditory system. When tested on the IBM TANGORA system, his front-end performed substantially better than a filter bank for speech recorded with a Crown PZM6A microphone.

While the use of models of the peripheral auditory system also provide a relative

[8]Discrete Fourier Transform.

improvement for the case of mismatches in training and testing over the baseline-processing case, the reasons for this are again not well understood. Also, these techniques are typically very computationally expensive. Although we believe that great insight can be obtained by analyzing the functioning of the human auditory system, we believe that other alternative approaches are likely to be more practical in the shorter term.

### 1.2.4. Techniques Based on Short-Time Spectral Amplitude Estimation

A number of algorithms have been developed to estimate a clean speech spectrum from the noisy speech spectrum. Spectral subtraction techniques were introduced by Boll [5] and Berouti *et al.* [3] in the context of speech enhancement. The noise spectrum was subtracted from the corrupted speech spectrum. While increasing the SNR, the speech intelligibility was not improved.

Recent implementations of spectral subtraction for speech recognition systems include the work of Van Compernolle [11] [10] and Stern and Acero [74]. Van Compernolle improved the robustness of the IBM speech recognition system when a desk-top microphone was used. The feature vector in the IBM system consists of the outputs of an auditory model. He proposed the use of channel equalization with a special implementation of spectral subtraction in the logarithm domain. The work of Stern and Acero was similar, although it was based on cepstral parameters derived from an LPC analysis.

Although *ad-hoc* techniques like spectral subtraction provide a moderate improvement, they are not derived with any optimum criterion. Porter and Boll [67] proposed to estimate DFT coefficients via MMSE[9] techniques, and this approach produced an improvement in performance over their original spectral-subtraction methods. They also showed that estimating the log-spectrum was more effective than estimating the spectrum directly as far as improving accuracy in the recognizer. The use of filter-bank outputs instead of DFT coefficients (Boll *et al.* [6]) resulted in even higher accuracy because of the smoothing effect across frequency.

The use of MMSE criteria provides more reliable estimates of the original speech signal at low SNR than the use of maximum likelihood estimation. These techniques are very effective, although they still suffer from the same plague that affects all approaches: at low SNR some of the processed speech does not represent legitimate speech.

---

[9]The minimum mean squared error criterion (MMSE) is used to derive estimates that minimize the expected squared error between the estimate and the random variable to be estimated.

### 1.2.5. Techniques Based on Mixture Densities

Some authors have very recently proposed to characterize the feature vectors input to speech recognizers by mixtures of Gaussian probability densities as a way of increasing noise robustness. Techniques based on short-time spectral amplitude estimation produced some frames at low SNRs that did not represent legitimate speech frames, because different frequency components were implicitly assumed to be independent of each other. From the plethora of density functions for which different components are not independent of each other, mixtures of Gaussian densities are a popular choice in the speech recognition community.

Ephraim *et al.* introduced MAP[10] [16] and MMSE estimates [17] for speech enhancement with the aid of HMMs. Both speech and noise were modeled as AR processes whose parameters were computed via an EM[11] algorithm that maximized the likelihood across the whole utterance. The enhanced signal exhibited a substantial increase in SNR, although the process was very computationally expensive because the time waveform was reconstructed for every iteration.

Nadas *et al.* [56] used a mixture model of the outputs of an ear model in the TANGORA IBM speech recognition system. In their algorithm, the MIXMAX labeler, an MAP approach was used to select the most likely mixture component given the noisy speech frame. An EM algorithm was used to maximize the sum of Gaussian densities until convergence was reached. The MIXMAX labeler was shown to increase the robustness of TANGORA significantly but, since they assumed that the corrupted speech at each frequency band was the maximum of the noise and the clean speech at that band, low energy speech frames were not modeled accurately.

Erell and Weintraub [19] also proposed an approach that was based on a mixture model for filter-bank outputs. They reported improved performance over techniques that did not incorporate correlation between frequency bands. Since the parameter vector in DECIPHER, the SRI system, was the cepstrum, they computed an MMSE estimate that weighted the contributions from all the mixtures, instead of labeling directly as Nadas did. Their model for the noise degradation was more elaborate than that used by Nadas, but their system did not have the system integration that Nadas' had because they had to cluster the acoustic space in both the spectral domain and cepstral domain.

---

[10] The maximum *a posteriori* criterion (MAP) selects the parameter that has the highest probability given the observed input. In this case, we select the most likely mixture component given the input frame.

[11] The EM, estimate-maximize, algorithm is an iterative algorithm to solve problems of maximum likelihood with incomplete data. The reader is referred to Appendix F for details.

Nadas *et al.* [57] proposed an adaptive labeling scheme that applied a time-varying transformation to the input speech frames in an effort to minimize the accumulated VQ[12] distortion. Since this approach is directed towards normalizing different transfer functions as well as noise levels, the adaptive labeling increased the robustness of TANGORA with different microphones and recording conditions. However, analysis of convergence is hard for a constantly drifting process.

Varga *et al.* [79] proposed an interesting approach in which instead of estimating the speech in noise, the HMMs were modified to allow for the presence of noise via noise masking. In [80] Varga and Moore actually used an HMM for the noise and another HMM for the speech. The Viterbi algorithm involved a search in a 3D space in this case, and non-stationary noises could be modeled. Although this idea is very powerful, their model for noise degradation was not very accurate, as it was based on many hard thresholds. Also, the computational complexity was very high.

Furui [24] investigated an unsupervised speaker adaptation method that was based on hierarchical spectral clustering of cepstral coefficients. The goal of the approach was to minimize the VQ distortion between the target acoustic space and a universal acoustic space. He suggested that this technique could also be used to increase the robustness of the system in noisy environments.

### 1.2.6. Other Techniques

There are a number of different approaches to the problem of speech recognition systems that are robust to noise. Some of these other techniques include neural networks and the use of microphone arrays.

Tamura and Waibel [76] suggested the use of a neural network for speech enhancement that was trained to minimize the difference between noisy and clean waveforms. In [77] Tamura and Nakamura used different affine transformations between hidden layers that corresponded to different phonemes. Neural network technology may be promising for robust speech recognition too, but for that they should operate on the transformed domain, spectrum or cepstrum, and not on the waveform.

Some efforts are being pursued in enhancing speech by using a microphone array (Flanagan *et al.* [22], Silverman [72], Van Compernolle [12]). The goal of this approach is to develop a directivity pattern so that noise sources arriving from a different angle than the desired speech are attenuated. While microphone arrays need to be explored in

[12]Vector Quantization (VQ) is a procedure that assigns each input frame to the closest prototype vector from a finite set. The measure of proximity in SPHINX is the Euclidean distance. See Chapter 2 for details.

the future, we believe that there is still room for improvement with a monophonic signal. In addition, these two approaches can complement each other.

While most previous efforts are geared towards suppressing stationary noise, some authors have focused on the difficult problem of speaker separation (Min *et al.* [53], Zelinski [82], Naylor and Boll [58]). These approaches work acceptably when the speech from both speakers is voiced and their pitch frequencies are far apart.

### 1.2.7. Discussion

One important factor in the selection of techniques to use was computational complexity, because we need to evaluate usefulness of speech input for personal computers. In this dissertation we opted not to work with auditory models nor iterative techniques based on AR models since they both present a high degree of computational complexity. Furthermore, although these techniques provide an increased degree of robustness over the case of no processing, they have not been shown to be superior to the other techniques described.

At the other end of the continuum in computational complexity are the weighted distortion measures. Unfortunately, we have found in pilot experiments that the distortion measures proposed in the literature provided essentially no improvement over the case of standard Euclidean distance for our database evaluated with SPHINX.

Since in this study we were concerned with the capabilities and limitations of techniques that used only one microphone, we did not investigate microphone-array techniques. We also did not address the issue of non-stationary noise either because we consider it an extremely difficult problem.

In this dissertation we will primarily explore approaches based on short-time spectral amplitude estimation and techniques that use mixture densities. They offer an attractive compromise between efficiency and accuracy.

## 1.3. Towards Environment-Independent Recognition

The goal of our research is to increase the robustness of the speech recognition systems with respect to changes in the environment. Since mismatches between training and testing conditions lead to a considerable degradation in performance, systems presently must be retrained for every different environment used. Even in the case of retraining, environments with a higher noise level suffer a loss of accuracy as compared to clean environments because some information is lost when noise is added. Going beyond that will require more sophisticated techniques.

A system can be labeled as *environment-independent* if the recognition accuracy for a new environment is the same or higher than that obtained when the system is retrained for that environment. Attaining such performance will be the goal of this dissertation.

In this section we will introduce the need for joint compensation of noise and equalization and an integrated approach for normalization in the cepstral domain.

### 1.3.1. Joint Compensation for Noise and Equalization

Most of the techniques described in the previous section to reduce the effects of mismatches between training and testing conditions deal only with additive noise. Nevertheless, we have also noted that *spectral tilt* can affect the performance of speech-recognition systems as well.

In this dissertation we develop a set of algorithms to accomplish the *joint* normalization of noise and spectral tilt. Although some efforts have previously been made to combinethe two types of processing (*e.g.* Van Compernolle [10], Erell and Weintraub [19]), both phenomena were treated independently. We will show that there is indeed an interaction between these phenomena and that further benefit is obtained if joint normalization is performed.

We will present several algorithms that adapt to new acoustical environments by estimating the noise and spectral tilt from input data. For this we use the concept of a universal acoustic space that is transformed to match the acoustic space of the current environment. We show that a few seconds of speech are sufficient to adapt to a different acoustical environment.

### 1.3.2. Processing in the Cepstral Domain: A Unified View

While successful noise-suppression algorithms operate in the frequency domain, most successful continuous speech recognizers operate in the cepstral domain. In this dissertation we will describe algorithms that perform the noise suppression in the cepstral domain, so that a larger degree of integration can be achieved for cepstral-based systems. In fact, the level of integration is so high that the algorithms can be implemented in an extremely efficient manner by a straight-forward modification of the vector quantizer module.

We show that compensation for noise and spectral tilt can be achieved by an additive correction in the cepstral domain. In addition to this bias compensation, a variance normalization is also possible. Finally, we propose a frequency normalization operation that can be expressed as a matrix multiplication operation.

We show in this dissertation that conditioning on the instantaneous SNR[13] is advantageous. We show that the instantaneous SNR captures a great deal of the information needed to perform the compensation.

Although frequency normalization may appear to be outside the scope of the problems of robustness to changes in the environment, we show that using the processing in the cepstral domain together with the concept of normalization of the acoustic space allow us to perform some adaptation to the long-term characteristics of the speaker. In this dissertation we present a novel method to accomplish this via adaptive warping of the frequency axis.

### 1.3.3. Measuring Performance Evaluation

Although many of the techniques described in the previous section are successful at some level, it has been difficult to compare them with each other because there does not exist a standard corpus for algorithm comparisons in the field of noise robustness like the one developed by DARPA for research in continuous speech recognition. The way many authors evaluate their algorithms is by comparing recognition accuracies for different noise levels for a system that has been trained with clean speech, with and without processing for robustness. While these measurements may demonstrate the effectiveness of a given method, they do not provide a basis for comparison with other authors' work because different authors use different tasks and noise levels. With a few exceptions, most authors do not consider the accuracy of the system when it is trained and tested on the noisy speech. We believe that this is an important benchmark for the evaluation of algorithms. To this end, we recorded a training database stereophonically using two different microphones: a close-talking microphone and a desk-top microphone.

Another common characterization of noisy speech databases is that of SNR. Although under some circumstances SNR can provide a good estimate of the degree of difficulty of a speech database that has been recorded in the presence of white noise, it does not characterize the database when the noise is colored or when the speech has been passed through a filter that has altered the frequency response, as is the case for real environments. We will use the simple average of speech and noise spectra as a characterization of a stereo database.

[13] We define the instantaneous SNR as the ratio of the short-time energy of the speech at a certain time over the average energy of the noise. For this work, we assume the noise to be stationary, so that the noise energy is supposed to be constant and can be estimated from previous frames.

## 1.4. Dissertation Outline

Chapter 2 describes the structure of the speech recognition system and the databases. Chapter 3 reviews some of the specifics of implementations of spectral subtraction and equalization, as well as other standard techniques described in the literature.

Chapter 4 presents the need for a joint compensation of noise and spectral tilt. Two algorithms will be presented: one in the frequency domain and the other in the cepstral domain. The latter introduces the idea of a normalization that depends exclusively on the instantaneous SNR of the input speech and is the basis for the *SNR-Dependent Cepstral Normalization* (SDCN) algorithm.

Chapter 5 introduces the *Codeword-Dependent Cepstral Normalization* (CDCN) algorithm to solve the problem of changes in the environment by using the idea of normalization of the acoustic space. CDCN also exhibits a better behavior for frames with low SNR.

Chapter 6 introduces several algorithms that build on SDCN and CDCN in an effort to obtain more accurate and efficient algorithms. We present the *Interpolated SDCN* algorithm and the *Fixed CDCN* algorithm as an evolution of the algorithms in Chapter 5.

In Chapter 7 we describe a method to perform frequency normalization within the context of cepstral processing in SPHINX. We use the bilinear transform to find the warping parameter of the frequency axis that minimizes the VQ distortion. Finally, Chapter 8 contains a summary of results, and Chapter 9 contains our conclusions and suggestions for future work.

# 2 Experimental Procedure

In this chapter we will give an overview of the SPHINX system, describe the database we used in this work, SNR characterizations, and the baseline results. It is important to note that in this study we were not concerned with elevating the absolute performance for a specific task but rather with comparing the relative merits of different algorithms.

## 2.1. An Overview of SPHINX

The SPHINX system was developed at CMU by Lee *et al.* [46]. It was the pioneer in speaker-independent large-vocabulary continuous-speech recognition, and it is still considered to be the most accurate system in that aspect. We will briefly describe the different blocks that compose the system, with greater emphasis in the signal processing and vector quantization, since these are aspects that this thesis concentrates on.

A block diagram of the early stages of SPHINX is shown in Figure 2-1.

### 2.1.1. Signal Processing

All speech recognition systems use a parametric representation rather than the waveform itself. Typically the parameters carry information on the envelope of the spectrum. SPHINX uses frequency-warped LPC cepstrum as its parameter set, that are computed as follows:

- Speech is digitized at a sampling rate of 16 kHz.
- A Hamming window of 320 samples (20 ms) is used every 10 ms.
- A preemphasis filter $H(z) = 1 - 0.97z^{-1}$ is applied.
- 14 Autocorrelation coefficients are computed.
- 14 LPC coefficients are derived from the Levinson-Durbin recursion.

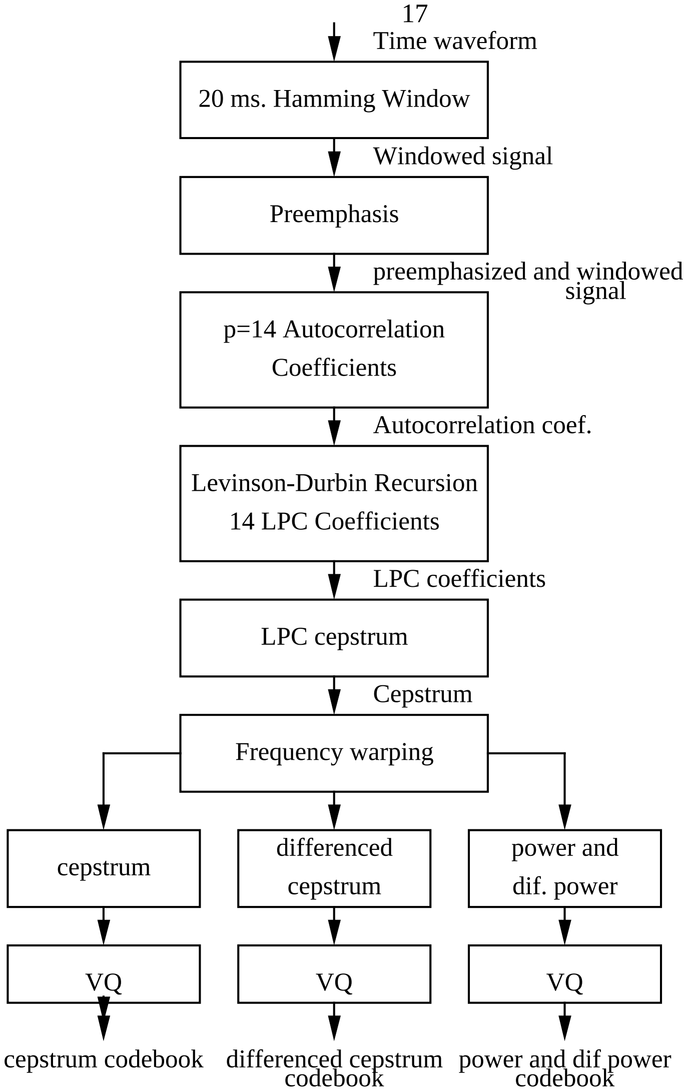



**Figure 2-1:** Block diagram of SPHINX front-end.

- 32 LPC cepstral coefficients are computed using the standard recursion
- These cepstral coefficients are frequency warped by using the bilinear transform producing 12 warped LPC cepstral coefficients.
- To account for differences in overall input level, the maximum value of the power $g$[14] in the utterance was subtracted for all frames so that after power normalization the maximum $g$ is 0.

Although adjacent frames are indeed correlated with each other, the SPHINX system assumes that every frame is statistically independent of the rest. In addition to the static

---

[14] $g$ is the zeroth order cepstral coefficient.

information provided by the cepstrum, SPHINX also uses dynamic information represented by the first order difference of cepstral vectors: $\mathbf{d}_i = \mathbf{x}_{i+2} - \mathbf{x}_{i-2}$. See Appendix B for a more detailed description of this processing.

### 2.1.2. Vector Quantization

Vector quantization (VQ) (Gray [31]) is a data reduction technique that maps a real vector onto a discrete symbol. Although it was originally proposed for speech coding, it has gained a great deal of popularity in speech recognition recently. A vector quantizer is defined by a codebook and a distortion measure:

- The discrete alphabet, called *codebook*, contains $L$ vectors, and it is a quantized representation of the vector space.
- The distortion measure estimates the degree of proximity of two vectors. An input vector is mapped to a symbol of this alphabet by choosing the *closest* codebook vector. In SPHINX the distortion measure used is the *Euclidean* distance.

SPHINX uses three different codebooks: one for the cepstrum, one for the first difference of cepstral vectors and the last one for power and the first difference of the power. By having each codebook contain 256 vectors, every frame of speech is condensed to 3 bytes. The distortion measure used is the Euclidean distance. The *prototype* vectors are estimated via a hierarchical clustering algorithm similar to the K-means algorithm developed by Linde *et al.* [50], which is an approximate maximum-likelihood method.

### 2.1.3. Hidden Markov Models

Hidden Markov Models (HMM), the dominant technology in continuous speech recognition, constitutes the recognition engine used in SPHINX. Rabiner and Juang [69] present a good review of HMMs, Picone [66] offers a summary of HMM applications to speech recognition.

Briefly, an HMM is a collection of states connected by transitions. Each transition carries two sets of probabilities:

- A transition probability which provides the probability for taking a transition from one state to the next, and
- An output probability density function (*pdf*), which defines the conditional probability of emitting each output symbol from a finite alphabet given that that transition is taken.

HMMs have become a widely-used approach for speech recognition due to the existence of maximum likelihood techniques to estimate the parameters of the models and algorithms that efficiently find the most likely state sequence.

### 2.1.4. Speech Units

Although the natural choice for a speech unit is the word, it is an impractical one due to the large number of words and the fact that numerous repetitions of each are needed to train the system adequately. A common speech unit is the phoneme, typically a 3-state left-to-right HMM, because it is easily trainable and because there are a small number of them (45 in an early version of SPHINX). Since the same phoneme in different contexts can be very different, SPHINX uses generalized *triphone* models as a way to model both the left and the right context.

Models for words are obtained by concatenating the models of the appropriate generalized triphones. It was found that using *function-word-dependent phones* helped improve recognition accuracy of function words such as *the, a, in, with* that occur frequently and are poorly articulated.

Finally the model for a complete utterance is obtained by a concatenation of the word models. A grammar can be incorporated in the HMM as another network whose nodes are words with different transition probabilities.

## 2.2. The Census Database

Although the bulk of research using the SPHINX system at Carnegie Mellon has made use of the well-known Resource Management database, we elected to use a different database, the census database, for our evaluations of signal processing. There are two reasons for this:

- The Resource Management database, with its large vocabulary size and many utterances, required about a week to train satisfactorily, which was excessively long since the entire system had to be retrained each time a new signal-processing algorithm was introduced.
- We specifically wanted to compare simultaneous recordings from close-talking and desk-top microphones in our evaluations. We believe that it is important to evaluate speech-recognition systems in the context of natural acoustical environments with natural noise sources, rather than using speech that is recorded in a quiet environment into which additive noise and spectral tilt are artificially injected.

We will now specify the speaker population, the database contents, the environment in which it was recorded and the recognition system we used to evaluate our algorithms.

### 2.2.1. Speaker Population

Since the system is to be speaker-independent, a large number of speakers are needed to train SPHINX. The census database has two disjoint segments:

- The training segment of the census database contains utterances from 74 speakers (53 male and 21 female).
- The testing segment contains utterances from 10 different speakers (7 male and 3 female) not present in the training.

All the speakers were selected from staff and students at CMU. The male/female ratio in the database reflects that of the general CMU population.

### 2.2.2. Database Contents

The database consisted of strings of letters, numbers, and a few control words, that were naturally elicited in the context of a task in which speakers spelled their names, addresses, and other personal information, and entered some random letter and digit strings. Specifically, each speaker read :

- 5 alphanumeric ("an") utterances that contained letters, digits and some control words (enter, erase, go, help, no, repeat, rubout, start, stop, yes) randomly. Some sample utterances are **N-S-V-H-6-T-49** and **ENTER-4-5-8-2-1**.
- 9 census ("cen") utterances containing respectively last name, first name, street number, street name, city, zip code, home phone number, birth date, and Social Security Number. Some sample utterances are **R-O-B-E-R-T** and **P-I-T-T-S-B-U-R-G-H**.

Since some of these utterances were discarded due to bad recordings, the total number of utterances for the training database is 1018. The testing segment of the database contains 140 utterances.

### 2.2.3. Alphanumeric Database

A total of 104 vocabulary items appeared in the vocabulary, of which 41 were uttered fewer than 10 times. No grammar was used in any of the experiments. This census task presents a greater degree of difficulty than the Resource Management task because:

- The perplexity[15] is larger. The census database with no grammar has a

[15]*Perplexity* is an information theoretic measure of the amount of constraint imposed by a finite-state grammar. If no grammar is used, the perplexity coincides with the size of the vocabulary. If a grammar is used to restrain the search space, the perplexity will be lower than the size of the vocabulary. In general higher perplexity tasks produce higher error rates.

perplexity of 104, while the perplexity of the word-pair grammar used in the resource management database is 60.

- The census vocabulary exhibits greater intrinsic acoustic confusability than the resource management vocabulary, because short words used in the census task such as the ones in the E-set (**B**, **C**, **D**, **E**, **G**, **P**, **T**) present a higher degree of acoustical similarity than the longer words in resource management.

### 2.2.4. The Environment

The census database was recorded simultaneously *in stereo* using both the Sennheiser HMD224 close-talking microphone that has been a standard in previous DARPA evaluations, and a desk-top Crown PZM6fs microphone. The recordings were made in one of the CMU speech laboratories (the "Agora" lab), which has high ceilings, concrete-block walls, and a carpeted floor. Although the recordings were made behind an acoustic partition, no attempt was made to silence other users of the room during recording sessions, and there is consequently a significant amount of audible interference from other talkers, key clicks from other workstations, slamming doors, and other sources of interference, as well as the reverberation from the room itself. Since the database was limited in size, it was necessary to perform repeated evaluations on the same test utterances.

### 2.2.5. The Recognition System

We also performed these evaluations using a more compact and easily-trained version of Sphinx with only 329 triphone models, omitting such features as duration, function-word and function-phrase models, between-word triphone models, and corrective training. We were willing to tolerate the somewhat lower absolute recognition accuracy that this version of Sphinx provided because of the reduced time required by the training process. Using the census database, the more compact Sphinx system, and DEC 3100 workstation, we were able to reduce the training time to the point that an entire train-and-test cycle could be performed in about 10 hours.

## 2.3. Objective Measurements

In this section we will summarize some definitions of Signal-to-Noise Ratio (SNR) given in the literature (See Jayant and Noll [40]) and present a new measurement that we believe provides more information about the environment.

Since the speech present in both microphone recordings was the same (as the database was recorded in stereo simultaneously), the main differences between them would be the noise levels and general tilt of the spectrum. We will see that conventional SNR figures of merit can only measure the difference in overall noise power, while a more detailed representation may be desired.

### 2.3.1. Measurements for Stationary Signals: SNR

SNR is defined as the ratio between the signal variance and the noise variance for a given signal $x[n]$. The noise variance is typically estimated as an average over a segment of a waveform that contains only noise samples.

If the speech database contains silent segments and we don't exclude them in the calculation of the signal power, the SNR figure will depend on how much silence our database contains, which is clearly undesirable for any objective measure. The fact that noise samples should be excluded in the computation brings up the issue of speech/silence discrimination.

The problem of discriminating speech and noise can be considered as a pattern recognition problem with two classes. Independently of what algorithm we use for this, the probability of labeling speech as noise and vice versa will be non-zero. The problem will become more severe as the noise level gets closer to the signal level, since the difference in means between the two distributions will be smaller.

In the case of our stereo database, the Sennheiser HMD224 close-talking microphone (CLSTK) exhibited an SNR of 38.4 dB whereas the Crown PZM6fs (CRPZM) had an SNR of 19.7 dB. A 20 ms Hamming window was used in the calculations.

### 2.3.2. Measurements for Nonstationary Signals: SEGSNR and MAXSNR

The above defined SNR is an adequate characterization of a channel if both signal and noise are stationary. For the time being we will be concerned with stationary noise, but it is well known that the speech signal is clearly *non-stationary*. By using the SNR as a measure, the high energy segments of the signal are dominating the computation. If this average is computed for a complete database, the overall SNR will be dominated by the speakers speaking most loudly.

To alleviate the fact that the speech signal is nonstationary and that the same amount of noise has different perceptual values depending on the ambient signal level, Noll [40] proposed the *segmental* SNR. The segmental SNR is based on a log-weighting that converts component SNR values to dB prior to averaging, so that very high SNR segments do not camouflage other segments with low SNR. A window has to be used in which speech is considered stationary (in our case a 20 ms Hamming window) as a basis for short-time integration. For our database the CLSTK and CRPZM exhibited a SEGSNR of 31.2 dB and 16.0 dB respectively.

Another measure proposed in the literature is SNRMAX, the maximum SNR in dB. This peak SNR has the advantage that it is not affected by the amount of silence the database has. On the other hand, just like when measuring maxima of *pdf*s, it is necessary to specify how the length of the window used (from one utterance to the whole database) and how the maximum is computed. We computed the SNRMAX as the average over all utterances in the test set of the maximum SNR. The results for the CLSTK and CRPZM were 50.3 dB and 29.6 dB respectively.

### 2.3.3. Frequency-Weighted SNR

This refinement recognizes that noise in certain frequency bands is less harmful than that in other bands of an input signal, and that signal in certain bands contributes more to intelligibility/recognition rate than in other bands. These measures have been extensively used in telephony:

- The *C-message weighting* function for speech represents the frequency response of the 500-type telephone set, as well as the hearing characteristics of the average telephone user.
- The *psophometric weighting* function, the European CCITT standard, is very similar to the C-message weighting function.
- For the *Articulation Index* used in the early speech work, the speech signal is observed in 20 sub-bands approximately distributed according to a Bark scale[16], with each band contributing equally to intelligibility. Component SNR values in dB (limited to a maximum allowable of 30 dB) are averaged:

$$AI = 0.05 \sum_{i=1}^{20} [\ \min\{SNR_i, 30\} / 30]$$

If, for example, the recordings include a 60 Hz hum, the SNR figure will be low although the recognition rate can still be high if those frequencies are filtered out. A measurement that would exclude those frequency bands would give a better estimate of the difficulty of the database from the standpoint of recognition accuracy.

The SPHINX system uses a technique similar to the *AI* based on frequency warping and a preemphasis filter that improve its accuracy. We argue that it is more meaningful to use a measure that uses this sort of frequency-weighted SNR in its calculation.

---

[16]The *Bark scale* is a warping of the frequency axis that is intended to better represent the frequency selectivity of the peripheral auditory system than a linear frequency scale.

### 2.3.4. A Proposed Solution: Average Speech and Noise Spectra

Since no single SNR measure will tell us anything about a tilt in the spectrum or about how the speech and noise energies are distributed across frequencies, we introduced a way of characterizing the databases that presents the results as a function of frequency.

To discriminate speech/non-speech frames we used a simple threshold on the zeroth cepstral component $g$, which is closely related to the energy of the frame. The minimum $g_{min}$ in the utterance was computed and all the frames below $g_{min}$ + 0.8 were considered noise while all the frames above $g_{min}$ + 1.5 were considered speech. The frames between these two thresholds are not included in the calculation. The cepstrum vectors of all frames classified as noise are averaged together, and so are the frames classified as speech. The DFTs of the resulting cepstral averages for the census database are plotted in Figure 2-2 for the Sennheiser HMD224 (CLSTK) and Crown PZM6fs (CRPZM), and the curves were normalized so that the noise level was 10 dB. The thresholds for classifying noise and speech were chosen empirically. The curves in Figure 2-2 computed by this method were not very sensitive to variations of the thresholds around the selected values. The classification done by this method was conservative in the sense that the system seldom misclassified speech as noise or vice versa. The price to pay was to have a region in which the frame was not classified as either noise or speech, but this was not important in this case as only an average is needed. In Chapter 6 we described a more elaborate method for noise estimation (Van Compernolle [11]) that performed similarly.

By comparing the curves in Figure 2-2, it can be seen that speech is about 25 dB above the noise level using the close-talking Sennheiser microphone. The signals from the Crown PZM, on the other hand, exhibit an SNR of less than 10 dB for frequencies below 1500 Hz and about 15 dB for frequencies above 2000 Hz. Furthermore, the response of the Crown PZM exhibits a greater spectral tilt than that of the Sennheiser, perhaps because the noise-canceling transducer on the Sennheiser also suppresses much of the low-frequency components of the speech signal. The so-called spectral tilt is the difference between the speech spectrum of the two microphone recordings.

The separation between curves in Figure 2-2 is smaller in magnitude than the SEGSNR, because these average spectra are computed by averaging the log-energies of all frequency bands rather than the energies themselves, as used in the SEGSNR calculation.

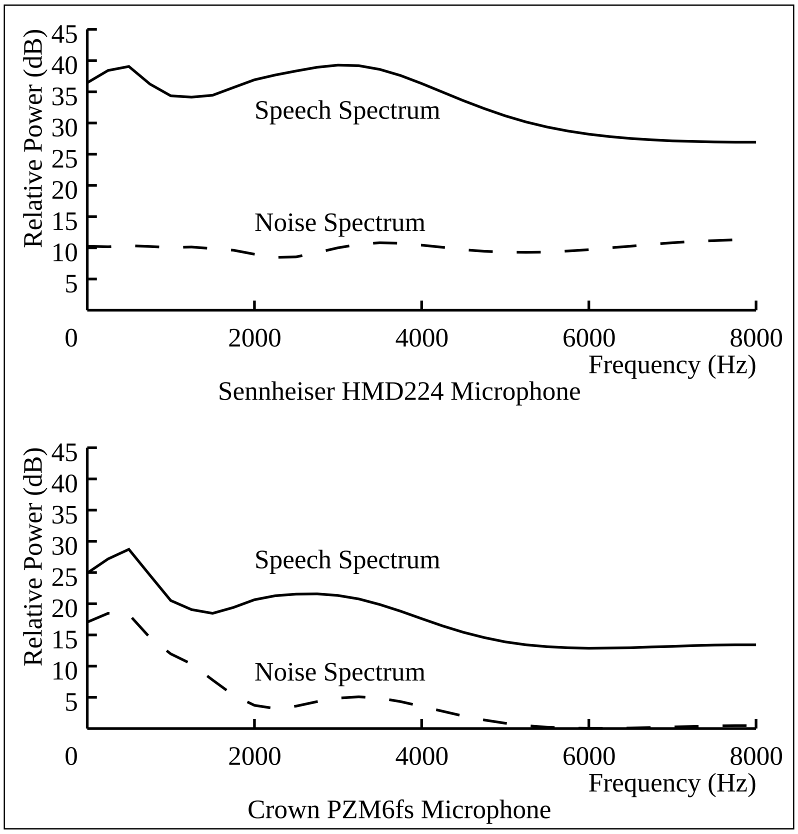



**Figure 2-2:** Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PZM6fs microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency. It can also be seen that the Crown PZM6sf produces greater spectral tilt.

### 2.3.5. Discussion of SNR Measures

In summary, the characterization of a speech database should not depend on aspects of the speech signal that are irrelevant such as

- Silent Periods and Background Noise. The objective measurement should reflect the *quality* of the recording, which does not depend on the length of the silent periods or background noise.
- Bandpass filtering. The level of an unfiltered speech signal mainly depends on the frequency components below 500 Hz while those frequencies only carry a part of the information. High-pass filters with cut-off frequencies of around 300 Hz are used in telephone communications.

We would expect that the acoustical characterization of some speech material will give some indication of how well a speech recognition system or a human can perform on it. In other words, the characterization should be correlated with the recognition rate and intelligibility. The use of average spectra provides a better indication of the difficulty of the database than measurements such as the SEGSNR because :

- It provides the SEGSNR for every frequency (noise in some frequencies is more harmful than in others), as opposed to averaging the SNR across all frequencies.
- It provides an indication of the tilt in the spectrum, which can degrade performance significantly, even for the same SEGSNR.

| | **CLSTK** | **CRPZM** |
|---|---|---|
| MAXSNR | 50.3 dB | 29.6 dB |
| SNR | 38.4 dB | 19.7 dB |
| SEGSNR | 31.2 dB | 16.0 dB |
| AVGSPT | 24.4 dB | 13.3 dB |

**Table 2-1:** Analysis of different SNR measures for the census database. In all cases a 20 ms Hamming window was used. All the figures are computed as the average in dB across all the utterances in the database of different measures: maximum signal energy in the utterance for MAXSNR, the average signal energy for SNR, the average log-energy of the signal for SEGSNR. AVGSPT is the average separation of curves in Figure 2-2

Table 2-1 shows the different SNR measures described for the census database. The differences between SNR, SEGSNR and AVGSPT are whether average of energies or log-energies are used across time and/or frequency.

## 2.4. Baseline Recognition Accuracy

We first consider the "baseline" recognition accuracy of the SPHINX system obtained using the two microphones with the standard signal processing routines. Table 2-2 summarizes the recognition accuracy obtained by training and testing using each of the two microphones. Recognition accuracy is reported using the standard DARPA scoring procedure reported by Pallett [62], with penalties for insertions and deletions as well as for substitutions. It can be seen that training and testing on the Crown PZM produces an error rate that is 60% worse than the error rate produced when the system is trained and tested on the Sennheiser microphone. When the system is trained using one microphone and tested using the other, however, the performance degrades to a very low level.

Hence we can identify two goals of signal processing for greater robustness: we need to drastically improve the performance of the system for the "cross conditions", and to elevate the absolute performance of the system when it is trained and tested using the Crown PZM.

| | Test **CLSTK** | Test **CRPZM** |
|---|---|---|
| Train **CLSTK** | 85.3 % | 18.6% |
| Train **CRPZM** | 36.9% | 76.5% |

**Table 2-2:** Baseline recognition rate of the Sphinx system when trained and tested on the census vocabulary using each of the two microphones.

### 2.4.1. Error Analysis

In order to better understand why performance degraded when the microphone was changed from the CLSTK to the CRPZM, even when the CRPZM was used for training as well as testing, we analyzed the errors that occurred in the CRPZM that did not occur in the CLSTK. We studied the spectrograms and listened carefully to all utterances for which training and testing with the CRPZM produced errors that did not appear when the system was trained and tested on the CLSTK. The estimated causes of the "new" errors using the CRPZM are summarized in Table 2-3. Not too surprisingly, the major consequence of using the CRPZM was that the effective SNR was lowered. As a result, there were many confusions of silence or noise segments with weak phonetic events. These confusions accounted for some 55 percent of the additional errors, with crosstalk (either by competing speakers or key clicks from other workstations) identified as the most significant other cause of new errors.

| **Type of error** | **Percent errors** |
|---|---|
| Weak-event insertion | 41.5 |
| Weak-event deletion | 13.2 |
| Crosstalk | 20.0 |
| Others | 25.3 |

**Table 2-3:** Analysis of causes of "new" errors introduced by use of the Crown PZM microphone.

## 2.5. Other Databases

Although the main database used in this work is the census database, we collected recordings using other microphones as well. In all cases the system was trained using the Sennheiser HMD224 (CLSTK). The "second" microphones (with which the system was *not* trained) were:

- The Crown PCC160 desk-top phase-coherent cardioid microphone (CRPCC160). (This is the new DARPA "standard" desk-top microphone.)
- An independent test set using the Crown PZM6fs.
- The Sennheiser 518 dynamic cardioid, hand-held microphone (SENN518).
- The Sennheiser ME80 electret supercardioid stand-mounted microphone (SENNME80).
- An HME lavalier microphone that also used an FM receiver (HME).

We recorded 140 utterances from 10 speakers for every microphone above and they were used in evaluating some algorithms after they had been developed.

We now summarize the salient acoustical attributes of these databases. The microphone recordings can be ranked according to decreasing SNR: the standard Sennheiser HMD224 (CLSTK), the Sennheiser 518, the Sennheiser ME80, HME FM, Crown PCC160 and Crown PZM6fs. Recordings with the Crown PCC160, Crown PZM6fs and HME FM exhibit considerable spectral tilt. A noticeable coloration of the noise spectrum is present for the Crown PZM6sf and the HME FM.

### 2.5.1. Sennheiser HMD224 - Crown PCC160

The average spectra for stereo recordings using the Sennheiser HMD224 and the Crown PCC160 are shown in Figure 2-3, and the SNR measurements are presented in Table 2-4. The baseline accuracy for the Sennheiser HMD224 and the Crown PCC160 were 82.4% and 70.2% respectively.

| | **CLSTK** | **CRPCC160** |
|---|---|---|
| MAXSNR | 52.8 dB | 36.1 dB |
| SNR | 41.2 dB | 26.0 dB |
| SEGSNR | 33.4 dB | 20.9 dB |

**Table 2-4:** Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Crown PCC160.

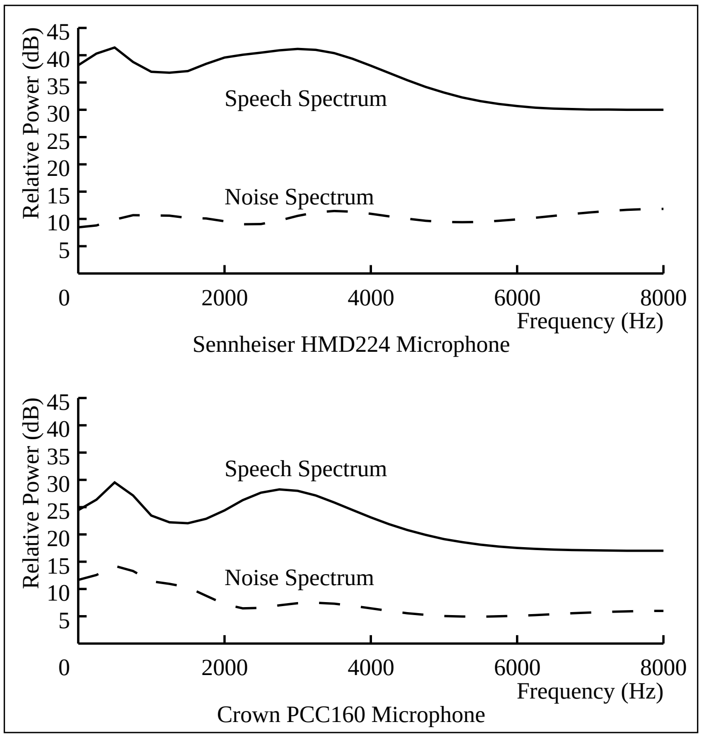



**Figure 2-3:** Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PCC160 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.

## 2.5.2. Sennheiser HMD224 - Crown PZM6fs

The average spectra for stereo recordings using the Sennheiser HMD224 and the Crown PZM6fs are shown in Figure 2-4, and the SNR measurements are presented in Table 2-5. The baseline accuracy for the Sennheiser HMD224 and the Crown PZM6fs were 84.8% and 41.8% respectively.

| | **CLSTK** | **CRPZM6fs** |
|---|---|---|
| MAXSNR | 53.0 dB | 33.4 dB |
| SNR | 41.5 dB | 23.4 dB |
| SEGSNR | 33.6 dB | 18.9 dB |

**Table 2-5:** Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Crown PZM6fs.

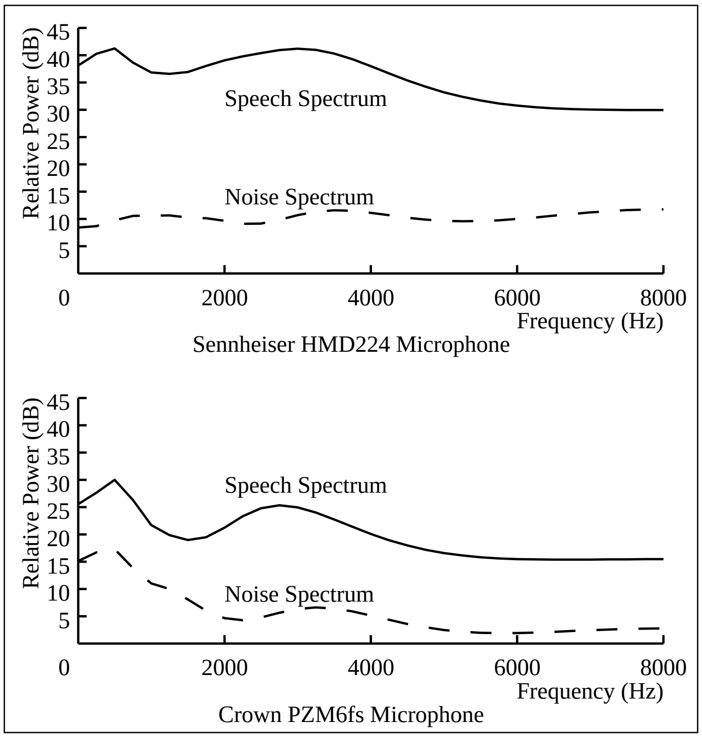



**Figure 2-4:** Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PZM6fs microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.

### 2.5.3. Sennheiser HMD224 - Sennheiser 518

The average spectra for stereo recordings using the Sennheiser HMD224 and the Sennheiser 518 are shown in Figure 2-5, and the SNR measurements are presented in Table 2-6. The baseline accuracy for the Sennheiser HMD224 and the Sennheiser 518 were 87.2% and 84.5% respectively.

| | **CLSTK** | **SENN518** |
|---|---|---|
| MAXSNR | 51.8 dB | 44.8 dB |
| SNR | 40.4 dB | 34.0 dB |
| SEGSNR | 32.6 dB | 27.4 dB |

**Table 2-6:** Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Sennheiser 518.

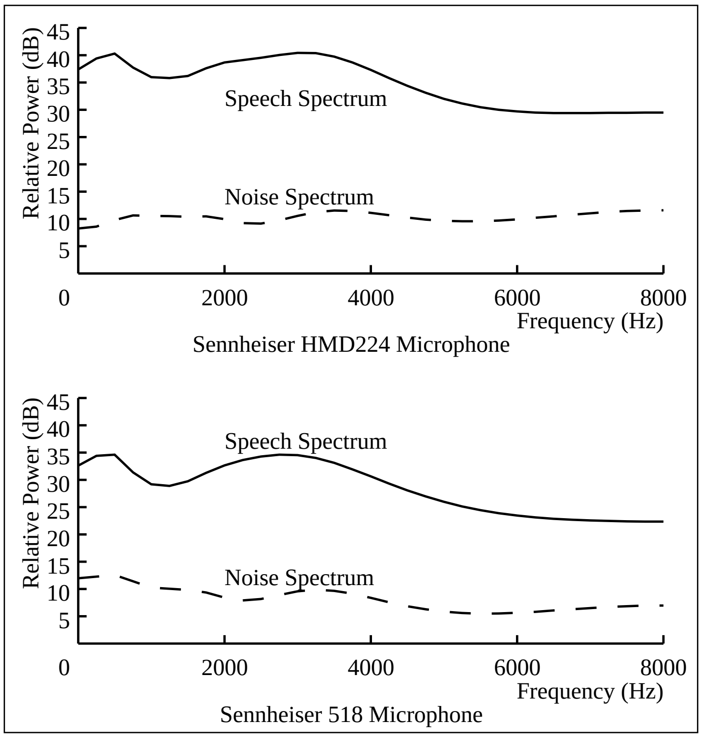



**Figure 2-5:** Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Sennheiser 518 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.

### 2.5.4. Sennheiser HMD224 - Sennheiser ME80

The average spectra for stereo recordings using the Sennheiser HMD224 and the Sennheiser ME80 are shown in Figure 2-6, and the SNR measurements are presented in Table 2-7. The baseline accuracy for the Sennheiser HMD224 and the Sennheiser ME80 were 83.7% and 71.4% respectively.

| | **CLSTK** | **SENNME80** |
|---|---|---|
| MAXSNR | 52.2 dB | 41.2 dB |
| SNR | 40.7 dB | 30.0 dB |
| SEGSNR | 33.0 dB | 23.2 dB |

**Table 2-7:** Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Sennheiser ME80.

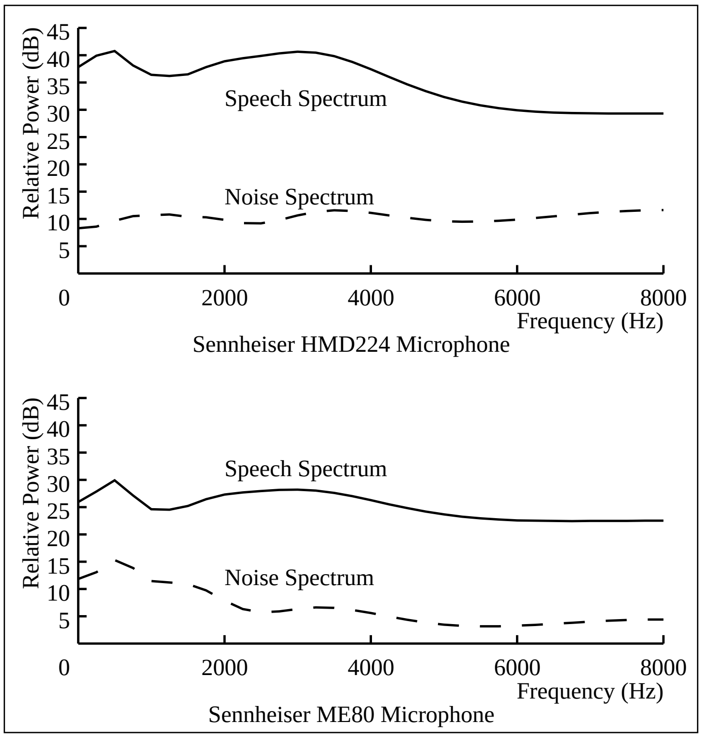



**Figure 2-6:** Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Sennheiser ME80 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.

### 2.5.5. HME FM - Crown PCC160

The average spectra for stereo recordings using the Sennheiser HMD224 and the HME FM are shown in Figure 2-7, and the SNR measurements are presented in Table 2-8. The baseline accuracy for the HME FM and the Crown PCC160 were 55.9% and 56.3% respectively.

| | **HMEFM** | **CRPCC160** |
|---|---|---|
| MAXSNR | 36.7 dB | 35.1 dB |
| SNR | 26.4 dB | 25.1 dB |
| SEGSNR | 21.2 dB | 20.1 dB |

**Table 2-8:** Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the HME FM and the Crown PCC160.

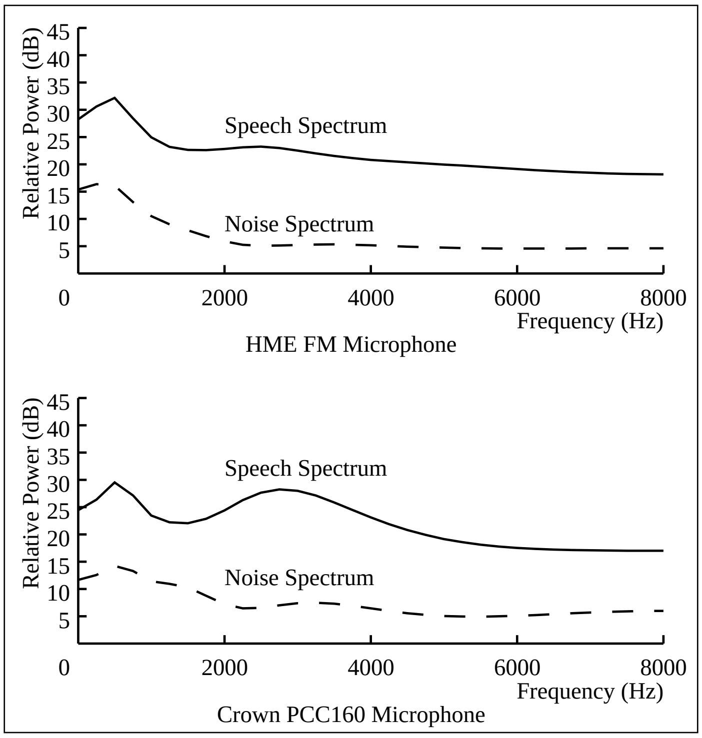



**Figure 2-7:** Average speech and noise spectra from the Alphanumeric database obtained using the HME FM Microphone and the Crown PCC160 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.

## 2.6. Summary

In this chapter a description of the experimental procedures to be used in the thesis is presented. We are fortunate to be able to use the state-of-the-art recognition engine. We described in more detail the front-end in SPHINX, and especially the cepstrum and VQ stages, since they will be central to the work in the rest of the dissertation.

After providing an overview of the SPHINX system, we proceeded to describe the census database for speaker-independent continuous speech that will be used in this thesis. The database was recorded in stereo simultaneously with two different microphones: a standard close-talking microphone and a desk-top one. This allow us to compare results in a more direct way.

Several possible characterizations of the database were proposed as a measure of its difficulty. While standard signal-to-noise ratio calculations provide a useful measurement, the proposed average spectra is a more detailed and useful characterization.

We then presented the baseline performance of the census database with SPHINX. The main results were that mismatches in training and testing conditions lead to a considerable degradation in performance. The motivation here is to increase the robustness of speech recognition systems that are tested under a number of different acoustical environments.

Finally, additional databases with different sets of microphones were recorded in stereo. The average spectra, SNR calculations and baseline performance were computed for all of them.

# 3 Processing in the Frequency Domain

In this chapter we will review some of the techniques proposed in the literature to deal with the problem of robustness to noise and tilt in the spectrum. These techniques were adapted to our present system and modified when necessary. Several approaches have been tried in conjunction with our census database and the SPHINX system. Specifically, we will describe the use of multi-style training in the case of multiple acoustical environments, and combinations of spectral subtraction and equalization algorithms.

## 3.1. Multi-Style Training

Multi-style training is a technique used in pattern recognition problems to increase robustness with respect to some variability or *styles*. Instead of using a model for that variability, the approach consists of including in the training a database that contains the variability.

This technique has been successfully used with Hidden Markov Models because of their powerful modeling abilities in different contexts. Lippmann *et al.* [51] pooled all the data from speakers speaking under different styles (fast, soft, angry, loud, Lombard[17]) to increase the robustness of the recognition system to different *speech styles*. Lee and Hon [45] showed that by including speech from a large number of different speakers in a training set, speaker independence can be achieved.

An experiment was carried out in which the system was trained on all the speech recorded from both the CLSTK and the CRPZM microphones (see Table 3-1). The multi-

[17]When the speaker is in an especially noisy environment, the produced speech, called *Lombard* speech, exhibits different characteristics than normal speech. This effect can also be simulated by recording speech from talkers in a quiet environment while they are listening to high-level noise presented through headphones.

style training is using twice as many training utterances, although they are not independent since they were recorded stereophonically. As expected, robustness is gained by using multi-style training but at the expense of sacrificing performance with respect to the case of train and test on the same conditions.

| | Test **CLSTK** | Test **CRPZM** |
|---|---|---|
| Train **CLSTK** | 85.3% | 18.6% |
| Train **CRPZM** | 36.9% | 76.5% |
| Multi-Style | 78.5% | 67.9% |

**Table 3-1:** Comparison of the baseline performance of the system under different training conditions: close-talking microphone, Crown PZM microphone and multi-style training. Testing is done for the two microphones.

By analyzing these results it is evident that greater robustness has been obtained by the multi-style training procedure. That is, the difference in performance between the two microphones is considerably smaller than it was under normal training conditions. Unfortunately we also see that we have to pay the price of having to tolerate lower recognition rates than the ones in which training and testing are done with the same microphone.

These results were not surprising as it has already been shown that although multi-style training increases the robustness of the recognition system, the accuracy is lower than the one obtained with training and testing on the same condition (speech style, speaker). Robustness to different speech styles is achieved at the expense of allowing some degradation in performance over the case of training and testing on the same conditions (Lippmann *et al.* [51]). Speaker independent systems typically exhibit an error rate that is 3 or 4 times as large as speaker dependent systems given that the same amount of training data is available in both cases (Pallett *et al.* [63]) for a given system.

To achieve *microphone-independent* recognition with multi-style training, recordings from more microphones and acoustical environments will be necessary. Lee and Hon [45] used 80 speakers to achieve *speaker independence* and it is not clear how many different acoustical environments would be necessary to obtain *microphone independence*. We can expect that as recordings from more microphones are used in training, a greater degradation in performance should be observed over the case of training and testing on the same acoustical environment.

Although multi-style training is a possible solution to microphone robustness, it would be desirable to attain the robustness without the penalty of a lower overall performance. We believe that any solution to a problem of robustness will almost always imply obtaining a good model for the observed degradations.

## 3.2. Channel Equalization

Spectral equalization is a filtering operation used to compensate for spectral tilt. Several researchers have used some kind of spectral equalization when dealing with particular problems. It has been applied in the following contexts:

- **Speaker Verification**. Atal [2] was the first author to propose some kind of long-term normalization for a speaker verification system. The average of the cepstrum vector throughout the utterance was subtracted from each individual vector in an attempt to reduce the possibility that an imposter could be misclassified. Furui [23] confirmed in another speaker-recognition experiment that subtracting a long-term average maintained a good recognition rate while providing robustness against channels with different frequency responses. Li and Porter [47] did some kind of speaker normalization for speaker verification.
- **Speech Styles**. Chen [7] has shown recently that the addition of a *fixed* mean vector to the cepstrum reduced the effects of different speech styles (soft, fast, shout, normal, Lombard).
- **Dereverberation**. Deconvolution in the cepstral domain has been used by Stockham *et al.* [75] to reduce the spectral tilt effect of low-quality recordings by Caruso caused by reverberation and resonances of the recording equipment.
- **Differences in microphone and recording conditions**. Morii and Stern [54] compensated for different spectral means between recordings done with a close-talking microphone and a desk-top one in speech recognition experiments by equalizing with a fixed transfer function[18]. They observed a significant improvement in performance, although having to return to the time domain after each operation made the strategy rather inefficient.

All these approaches have in common the removal of a mean spectral vector and in all cases that mean vector was the sample mean. While some success has been achieved in specific domains of application, little integrating effort has been done to combine different sources of variability into environment independent speech recognition.

[18]The transfer function of the equalizer was simply the ratio between the average power spectral density functions obtained for the two sets of recordings. The filtering was carried out by a method similar to the OLA (overlap and add) method.

## 3.3. Noise Suppression by Spectral Subtraction

Spectral subtraction refers to a family of techniques designed to suppress or reduce the noise in a signal. Although it was originally proposed in the context of speech enhancement (Boll [5], Berouti *et al.* [3]), there has been a great deal of recent interest in its application to robust speech recognition. In the latter case, the end user of the processed speech is not a human being but a computer.

We present in this section an introduction to spectral subtraction with its use in speech enhancement and recognition. Then a framework for processing in the log-spectrum is presented.

### 3.3.1. Spectral Subtraction for Speech Enhancement

Spectral subtraction is a family of techniques that attempt to subtract the noise energy from the noisy speech energy at every frequency band. It can be formalized mathematically by assuming that the speech signal $x[m]$ is corrupted by additive noise $n[m]$ that is uncorrelated with $x[m]$[19]:

$$y[m] = x[m] + n[m] \tag{3.1}$$

Let's define $Y(\omega)$, $X(\omega)$, and $N(\omega)$ as the power spectral densities (PSD) of the signals $y[m]$, $x[m]$ and $n[m]$ respectively. Since the signal and the noise are uncorrelated with each other, the following relationship holds for every frequency band $\omega_k$:

$$Y(\omega_k) = X(\omega_k) + N(\omega_k) \tag{3.2}$$

If we obtain an estimate of the PSD of the corrupted signal $\hat{Y}(\omega)$ at a certain frame, and an estimate of the PSD of the noise $\hat{N}(\omega)$ from regions where no speech is present, we could obtain an estimate of the PSD of the desired signal as

$$\hat{X}(\omega_k) = \hat{Y}(\omega_k) - \hat{N}(\omega_k) \tag{3.3}$$

and thus the name of spectral subtraction. Since this estimate $\hat{X}(\omega_k)$ can go negative, Equation (3.3) is modified to disallow it by means of a half-wave rectification:

$$\hat{X}(\omega_k) = \max\{\hat{Y}(\omega_k) - \hat{N}(\omega_k), 0\} \tag{3.4}$$

We now show that under the assumption that the power spectrum is normally distributed, the ML estimate for the undegraded signal yields (3.4), the basic spectral subtraction rule. We can obtain an expression for the *joint pdf* as a product of the individual *pdfs*:

---

[19]All notational conventions are summarized in Appendix A

$$p(\hat{Y}(\omega_k), \hat{N}(\omega_k) \mid Y(\omega_k), N(\omega_k))$$
$$= p(\hat{Y}(\omega_k) \mid Y(\omega_k), N(\omega_k), \hat{N}(\omega_k))\, p(\hat{N}(\omega_k) \mid Y(\omega_k), N(\omega_k))$$
$$= p(\hat{Y}(\omega_k) \mid Y(\omega_k))\, p(\hat{N}(\omega_k) \mid N(\omega_k)) \tag{3.5}$$

where we have used the facts that the estimate of the corrupted signal depends only on the corrupted signal itself, and not on the noise nor its statistics. Likewise, the estimate of the noise is assumed to be independent of the corrupted signal (by doing the estimation when no signal is present for instance). By using the Gaussian assumption, the likelihood will be expressed as

$$l(\hat{Y}(\omega_k), \hat{N}(\omega_k) \mid Y(\omega_k), N(\omega_k)) = -\ln\{2\pi\sigma_y\sigma_n\}$$
$$-\frac{(\hat{Y}(\omega_k) - Y(\omega_k))^2}{2\sigma_y^2} - \frac{(\hat{N}(\omega_k) - N(\omega_k))^2}{2\sigma_n^2} \tag{3.6}$$

whose maximization clearly yields $\hat{Y}(\omega_k) = Y(\omega_k)$, $\hat{N}(\omega_k) = N(\omega_k)$ if $Y(\omega_k) > N(\omega_k)$ and hence $\hat{X}(\omega_k) = \hat{Y}(\omega_k) - \hat{N}(\omega_k)$.

The idea of spectral subtraction was originally proposed by Boll [5] in the context of speech enhancement. The spectrum of the restored signal had the same phase as that of the corrupted signal, and a magnitude that was set equal to the difference between the magnitude spectra of $y[m]$ and $n[m]$. All spectra were estimated via DFTs. Boll used overlapping Hanning[20] windows and the overlap-and-add method to obtain the restored signal. His approach differs from the basic concept described above in that magnitude subtraction rather than power subtraction was used. Since speech recognition systems don't use the phase information, we will not be concerned with that issue here. The noise spectrum was estimated by averaging the magnitude of several noise frames. At low SNR the processed speech exhibited a residual noise, characterized by random spikes. He proposed additional residual noise reduction schemes, especially during non-speech activity.

Berouti *et al.* [3] proposed a modified version of the power subtraction rule (3.4) in which the amount of noise subtraction depended on the SNR of the particular frame. The oversubtraction was done to combat the residual noise that Berouti called *musical noise*. The enhanced speech obtained by these methods exhibited a greater SNR, although this processing didn't increase the intelligibility.

---

[20]The *Hanning* window is a raised cosine $h[m] = 1 - \cos(2\pi m/(N-1))$ for $0 \le m \le N-1$ and 0 otherwise.

### 3.3.2. Noise Subtraction in Speech Recognition

The recent use of spectral subtraction in ASR has proved more successful than in the arena of speech enhancement. Morii [54] showed in pilot experiments with SPHINX that the accuracy of the system did not deteriorate as rapidly if the waveform was enhanced by spectral subtraction techniques. Since most speech recognition systems use some kind of spectral representation as the feature space, it is unnecessary to recreate the speech signal, and one should do the subtraction directly in the transformed domain.

Porter and Boll [67] proposed MMSE estimators of DFT bins for use in speech recognition. In their work, the authors proposed a statistical characterization by assuming that the real and imaginary parts of the Fourier transform are independent and Gaussian distributed. Under these assumptions it can be shown that the magnitude of the spectrum exhibits a Rayleigh distribution. An MMSE estimator is defined for a compression function of the magnitude spectrum as the non-linearity that minimizes:

$$E\{[f_c(\hat{X}(\omega_k)) - f_c(X(\omega_k))]^2 \mid Y(\omega_k), N(\omega_k)\} \tag{3.7}$$

whose solution $c(\hat{S}(\omega_k))$ equals

$$\frac{\int c(X(\omega_k)) \exp(-X(\omega_k)/N(\omega_k))\, I_0(2\sqrt{Y(\omega_k)X(\omega_k)}/N(\omega_k))\; p(X(\omega_k))\, dX(\omega_k)}{\int \exp(-X(\omega_k)/N(\omega_k))\, I_0(2\sqrt{Y(\omega_k)X(\omega_k)}/N(\omega_k))\; p(X(\omega_k))\, dX(\omega_k)} \tag{3.8}$$

where $I_0(x)$ is the zero$^{th}$ order Bessel function. Instead of assuming a model for the distribution of speech $p(X(\omega_k))$, the integrals were approximated by a sum for all the samples of speech within a specified database. Porter and Boll tried several compression functions $f_c()$, and among them the logarithm was the one that provided the highest recognition accuracy. However, even after processing the error rate was still 5 times the error rate for clean speech.

Ephraim and Malah [14] used the same Equation (3.8) but since they assumed a Gaussian *a priori* density for speech, they were able to get a closed-form expression for (3.8), although quite complicated. Ephraim and Malah [15] also claimed that greater enhancement would be obtained if the logarithm was used as a compression function.

### 3.3.3. Spectral Subtraction in the Logarithm Domain

It is useful to work in the logarithm domain because distortion measures that operate in the log-spectrum have been shown to work better than the ones that operate with the regular spectrum (Gray *et al.* [30]). Boll *et al.* [6] applied the MMSE estimator he derived for DFT bins in [67] to the log-amplitudes of the outputs of a *mel-scale* filterbank

[21] and obtained a greater degree of robustness. The differences with his previous approach (Porter and Boll [67]) were that the logarithm of the spectrum was used instead of the magnitude, and that filterbank outputs rather than DFT coefficients were used. The difference in recognition accuracy between the MMSE technique and Boll's own spectral subtraction technique was not as dramatic as it was when the magnitude of DFT bins were used. Two things can be learned from this experience:

- Filterbank outputs are preferred to DFT coefficients since they provide better performance. Although using different databases, the work by Boll [5] [6] indicates that the smoothing effect across frequency provided by the filterbank is very beneficial in reducing the variance of the estimate.
- Measures in the logarithm domain are more robust for recognition and they are also more consistent with the processing done in the VQ stage of SPHINX.

These results, together with the fact that a large number of speech recognition systems choose some kind of cepstral representation (in the logarithm domain), led us to investigate how the spectral subtraction rule would be in the logarithm domain.

The logarithms of the power spectral densities of the noisy speech $Y(\omega_k)$, the clean speech $X(\omega_k)$ and the noise $N(\omega_k)$ are represented by $\mathbf{Y}(\omega_k)$, $\mathbf{X}(\omega_k)$ and $\mathbf{N}(\omega_k)$ respectively. Under this notation the relationship $Y(\omega_k) = X(\omega_k) + N(\omega_k)$ can be expressed as

$$\begin{aligned}\mathbf{Y}(\omega_k) &= \ln Y(\omega_k) = \ln(X(\omega_k) + N(\omega_k)) = \ln N(\omega_k) + \ln\left(\frac{X(\omega_k)}{N(\omega_k)} + 1\right)\\ &= \mathbf{N}(\omega_k) + \ln(\exp(\mathbf{X}(\omega_k) - \mathbf{N}(\omega_k)) + 1)\end{aligned} \tag{3.9}$$

Let us further define the normalized values $\overline{\mathbf{Y}}(\omega_k) = \mathbf{Y}(\omega_k) - \mathbf{N}(\omega_k)$ and $\overline{\mathbf{X}}(\omega_k) = \mathbf{X}(\omega_k) - \mathbf{N}(\omega_k)$, that have the interpretation of channel SNR. Now the power addition in the logarithm domain is given by $\overline{\mathbf{Y}}(\omega_k) = \ln(\exp(\overline{\mathbf{X}}(\omega_k)) + 1)$. Similarly we can derive the power subtraction in the logarithm domain as $\overline{\mathbf{X}}(\omega_k) = \ln(\exp(\overline{\mathbf{Y}}(\omega_k)) - 1)$. Of course this expression is only defined for $\overline{\mathbf{Y}}(\omega_k) > 0$ or equivalently $Y(\omega_k) > N(\omega_k)$. The power spectral subtraction rule (3.4) can be translated into the logarithm domain as $\overline{\mathbf{S}}(\omega_k) = \max\{\ln(\exp(\overline{\mathbf{Y}}(\omega_k)) - 1), -\infty\}$.

It is clear by examining the spectral subtraction rule in the logarithm domain that

[21] A mel-scale filterbank has the center frequencies of the filters spaced non-linearly approximating the peripheral auditory system.

making $\overline{\mathbf{X}}(\omega_k) = -\infty$ will yield an infinite value for VQ distortions that will in turn result in errors in the recognizer. The $-\infty$ appears because we have implicitly assumed that the original signal was completely noise-free, which is not realistic. Since our clean speech will still have some noise (quantization, aliasing, estimation error, etc) the SNR will be high but not infinite. Our goal is to increase the SNR of the corrupted signal to the level it was before.

If this *a priori* information is taken into account, the spectral subtraction rule in the logarithm domain will have the form

$$\overline{\mathbf{X}}(\omega_k) = \max\{\ln(\exp(\overline{\mathbf{Y}}(\omega_k)) - 1), \mathbf{X}_{th}(\omega_k) - \hat{\mathbf{N}}(\omega_k)\} \tag{3.10}$$

where $\hat{\mathbf{N}}(\omega_k)$ is the estimate of the noise level and $\mathbf{X}_{th}(\omega_k)$ is the signal floor level. This new spectral subtraction will take the input log-power $\mathbf{Y}(\omega_k)$ and produce an estimate of the log-power of the clean signal $\mathbf{X}(\omega_k)$ given by $\hat{\mathbf{X}}(\omega_k) = \max\{\mathbf{X}_{th}(\omega_k), \ln(\exp(\hat{\mathbf{Y}}(\omega_k)) - \exp(\hat{\mathbf{N}}(\omega_k)))\}$. Figure 3-1 shows a typical curve that can be implemented as a table lookup. Van Compernolle [10] also proposed the same transformation.

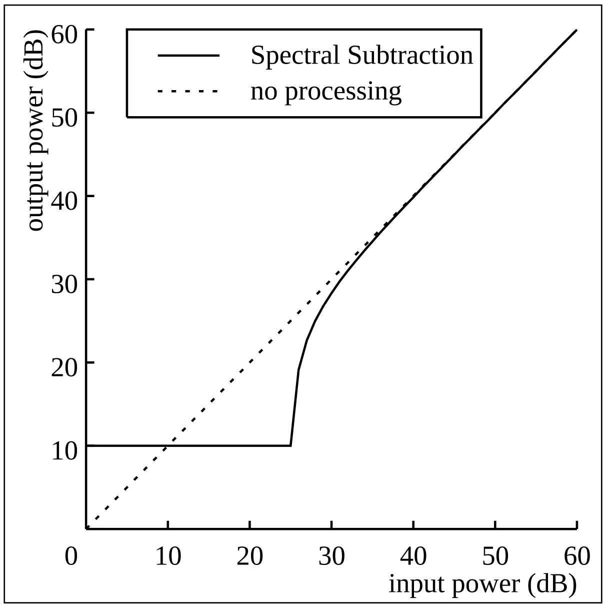



**Figure 3-1:** Spectral subtraction curve for $\hat{\mathbf{N}}(\omega_k) = 25\,\mathrm{dB}$ and $\mathbf{X}_{th}(\omega_k) = 10\,\mathrm{dB}$.

## 3.4. Experiments with Sphinx

In this section we describe a series of experiments performed using the census database, running the SPHINX system with a number of spectral equalization and noise suppression algorithms. In spectral subtraction and equalization it is assumed that the speech signal $x[m]$ is degraded by linear filtering and/or uncorrelated additive noise, as depicted in Figure 3-2. The goal of the compensation is to reverse the effects of these degradations.

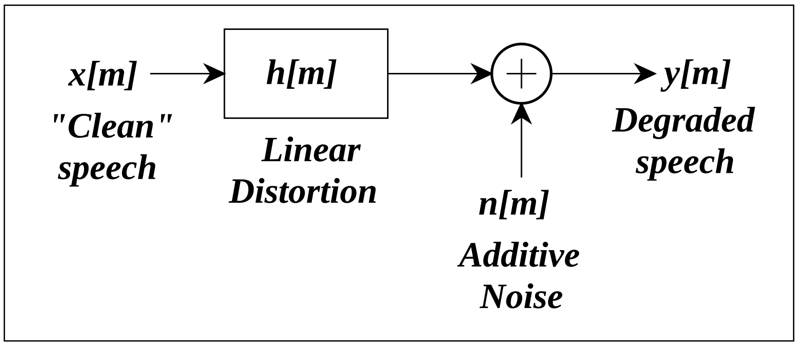



**Figure 3-2:** Model of the degradation.

Spectral equalization techniques attempt to compensate for the filter $h[m]$, while spectral subtraction techniques attempt to remove the effects of the noise from the signal. We compare the performance of several different implementations of spectral subtraction and equalization techniques in Table 3-7, which we refer as EQUAL, PSUB, MMSE1 and MSUB. These algorithms were applied *only* to the CRPZM speech, as the goal of the compensation schemes is to be able to use the HMM models trained with the CLSTK. In this section we describe these algorithms and examine the extent to which they make the SPHINX system more environmentally robust.

To illustrate their performance, we will consider three-dimensional spectrograms of one sample utterance for all algorithms. In Figures 3-3 and 3-4 we show 3-D plots of the word *yes* recorded with the CLSTK and the CRPZM microphone respectively. These 3-D plots are a time-frequency representation of the signal obtained by taking the DFT of the cepstral vectors computed by SPHINX's front-end. The frequency axis is warped by the bilinear transform (See Chapter 7). Since VQ labels reflect different spectral shapes, the labels assigned to the noise segment of the CRPZM (up to 330 ms, and beyond 750 ms) are very different than the ones assigned for the CLSTK. This is a major cause for the errors observed in Table 3-2.

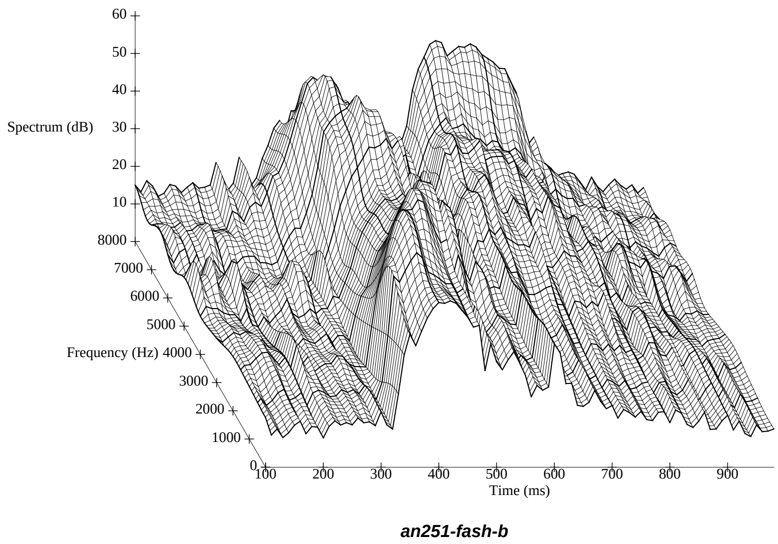



**Figure 3-3:** 3D Spectrogram of the utterance *yes* recorded with the CLSTK microphone with no processing.

### 3.4.1. EQUAL Algorithm

EQUAL is a spectral equalization algorithm that compensates for the effects of the linear filtering, but not the additive noise, and was first described in [74] (Stern and Acero). This algorithm is inspired by work done by Stockham *et al.* [75] in blind deconvolution. Compensation for linear convolution is accomplished by multiplying the spectrum of the input signal by the spectrum of the compensating filter[22]. Therefore the cleaned log-spectrum $\mathbf{X}(\omega_k)$ can be estimated from the input log-spectrum $\mathbf{Y}(\omega_k)$ and the equalization $\mathbf{Q}(\omega_k)$:

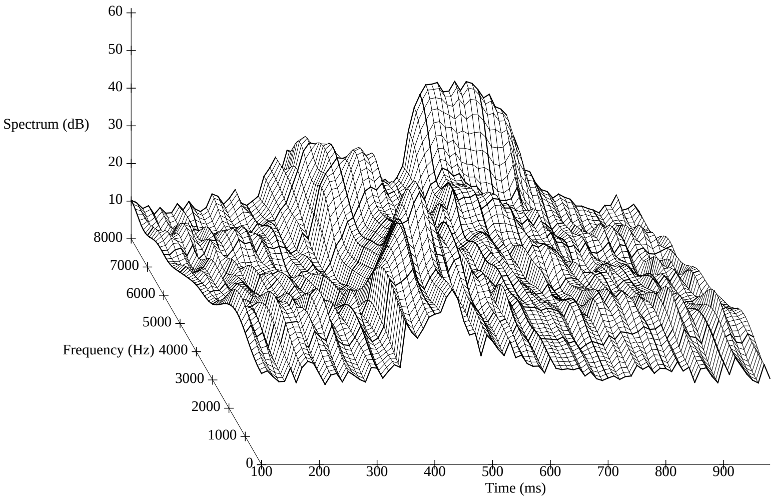
$$\hat{\mathbf{X}}(\omega_k) = \mathbf{Y}(\omega_k) + \mathbf{Q}(\omega_k) \tag{3.11}$$

The equalization $\mathbf{Q}(\omega_k)$ is estimated as the difference between the average log-spectra for the two microphones, computed during speech activity (an approximate MMSE estimate, unless all the frames in the calculation can be considered noise free). Alternatively, the compensation can be applied as an additive correction in the cepstral domain to the CRPZM speech.

In Figure 3-5 we can see the spectrum of the sample utterance for the CRPZM after processing by this algorithm. By comparing the processed frame at time 400 ms of Figure 3-5 with that of Figure 3-4, we see that some spectral tilt has been removed. There is still

---

[22]Since the cepstrum is the inverse Fourier transform of the logarithm of the magnitude of the spectrum, and we are not interested in the phase for recognition, this corresponds to a simple additive correction of the cepstrum vector.




**an251-fash-c**

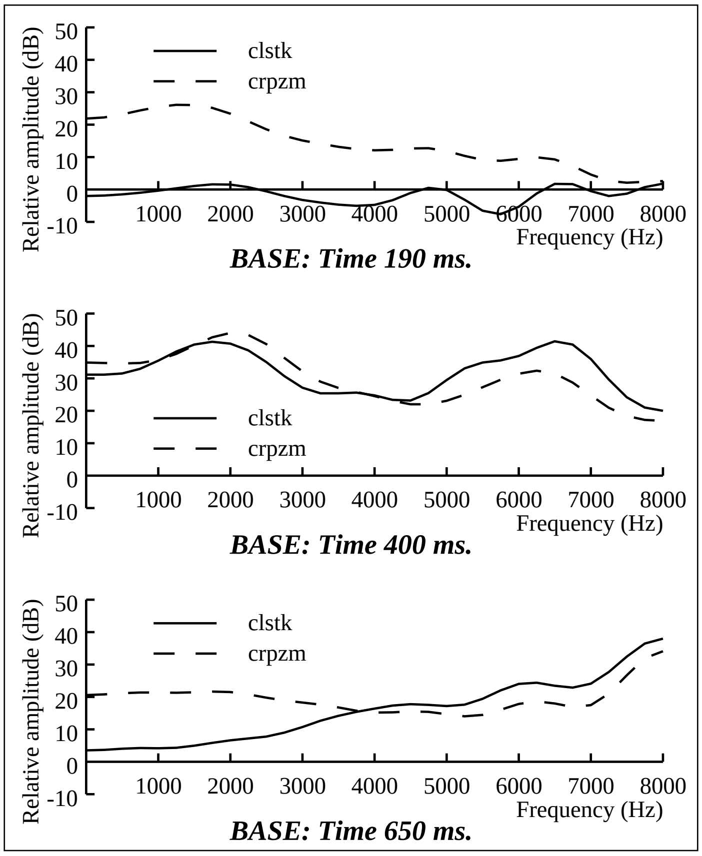



**Figure 3-4:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with no processing. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |

**Table 3-2:** Baseline recognition accuracy for the CLSTK and CRPZM.

some residual spectral tilt, because a single equalization function is used for all the speakers and utterances, whereas a different one would be needed (The spectral tilt varies from speaker to speaker, due to changes in the relative location of the mouth and the microphone.). The results in Table 3-3, show some improvement in performance for the cross conditions over the baseline case.

### 3.4.2. PSUB Algorithm

PSUB is an implementation of the power spectral subtraction rule in the logarithm domain (3.10). The transformation is accomplished by converting the cepstral coefficients to log-power coefficients using a 32-point inverse DFT (for the 32 real and even cepstral coefficients), performing the subtraction in these 32 frequency bands and converting once again to the cepstral domain by another DFT.

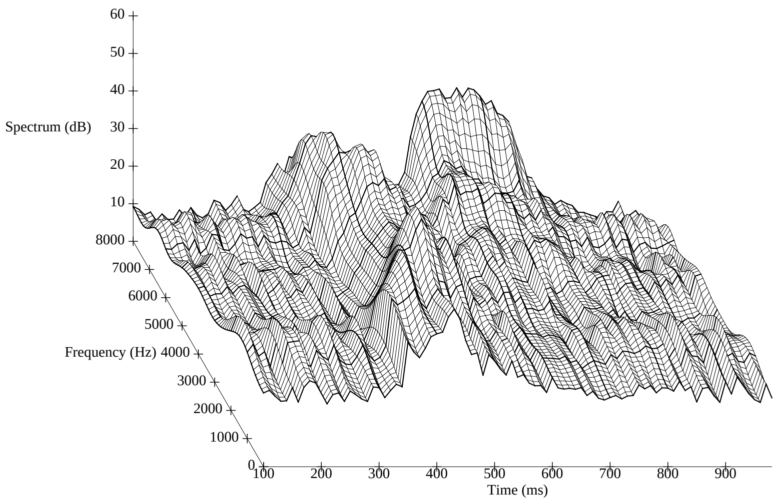
The estimate of the noise is computed by averaging the cepstrum vectors whose total energy fell under a threshold for *g*[23]. In our case this threshold is empirically set to the minimum *g* in the utterance plus 1.0, although the performance was not very sensitive to small variations of this value.

Finally, we note that the noise floor was picked in such a way that the dynamic range for the utterance in *g* was always 10.0 (equivalent to 44 dB). This was the optimum value as found by experimentation. We have found that the recognition accuracy was not very sensitive to this level.

In Figure 3-6 we see the spectrum of the example utterance for the CRPZM after processing by this algorithm, including the presence at low SNR of what Berouti *et al.* [3] called *musical noise*. Although the mean has been lowered, the residual noise exhibits spikes uncharacteristic of speech frames (See frames at time 190 ms and 650 ms). Since the vector quantizer essentially clusters spectral shapes, these spikes will cause large errors in the VQ labeler that in turn will translate into errors in the recognizer. Nevertheless, the evaluation results in Table 3-4 show some additional improvement in

---

[23] In this case speech/noise discrimination was done on a frame-by-frame basis, assuming that different speech frames were statistically independent. Furthermore, since only *g* was used for the classification, the optimum scheme is a comparison with a threshold.



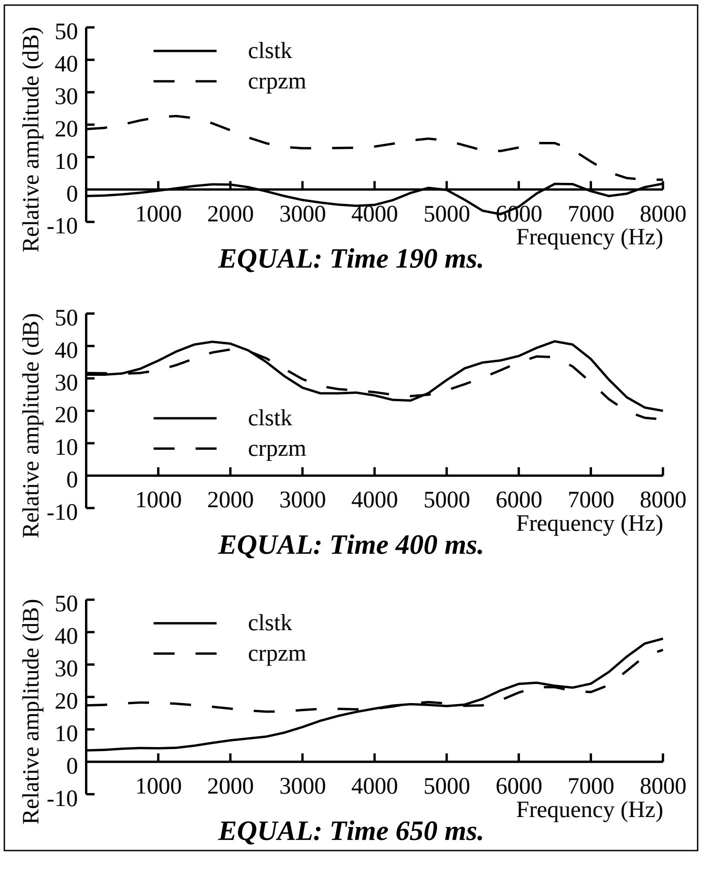




**Figure 3-5:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with EQUAL algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **EQUAL** | N/A | 38.3% | 50.9% | 76.5% |

**Table 3-3:** Comparison of the baseline performance and the EQUAL spectral equalization algorithm. EQUAL was only applied to the CRPZM.

performance over the baseline condition when the PSUB algorithm is applied to the CRPZM speech.

### 3.4.3. MSUB Algorithm

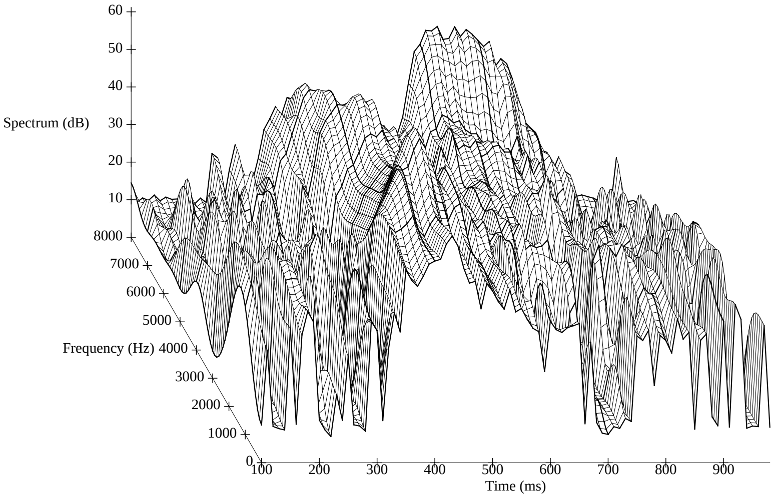
To ameliorate the problem of the *musical noise* present in the PSUB algorithm we developed the MSUB algorithm, first described by Stern and Acero in [74]. We introduced over-subtraction at low SNR to further suppress the *musical noise* following an idea originally suggested by Berouti *et al.* [3]. We note that we subtract the magnitudes of spectra as did Boll [5] rather than the more intuitively appealing spectral power because we found that magnitude subtraction provided greater recognition accuracy. The amount of over-subtraction or under-subtraction is a function of the instantaneous SNR according to the transformation shown in Figure 3-7. This transformation was developed empirically by analyzing 3-D spectrograms obtained with the MSUB algorithm.

In Figure 3-8 we show the spectrum of the example utterance for the CRPZM after processing by the MSUB algorithm. We note that MSUB suppresses the musical noise present in the PSUB algorithm (See frames at time 190 ms and 650 ms). The evaluation results in Table 3-5 show the effectiveness of using the magnitude rather than the power, and the use of the overall instantaneous SNR in determining under and over subtraction. The MSUB algorithm performs considerably better than the PSUB algorithm.

### 3.4.4. MMSE1 Algorithm

While the vector quantizer uses an MMSE criterion, the noise suppression algorithms derived so far are not goal-directed. Perhaps overall system performance could be improved if both modules would use the same MMSE criterion. Porter and Boll [67] realized this fact and used an MMSE estimator based on Equations (3.7) and (3.8). He suggested the use of the logarithm as a compression function. We will follow a similar approach here.

Let $\mathbf{N}(\omega_k)$ be the log-energy of the noise in band $k$, and $\mathbf{Y}(\omega_k)$ the log-energy of the corrupted signal in band $k$. We define the SNR in band $k$ or normalized input $\overline{\mathbf{Y}}(\omega_k)$ as:




*an251-psub-c*

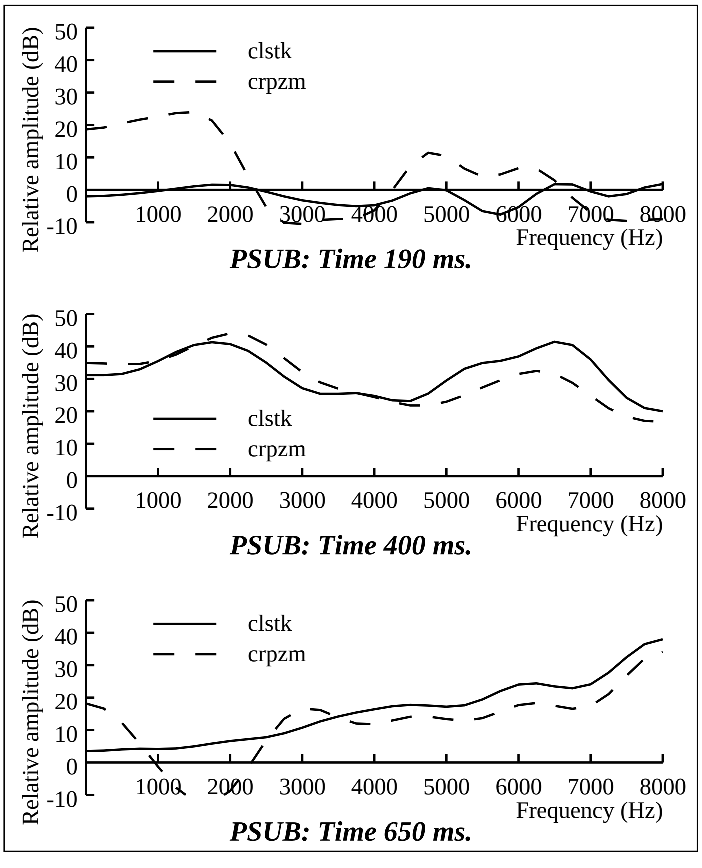



**Figure 3-6:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with PSUB algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **PSUB** | N/A | 38.6% | 70.6% | 70.1% |

**Table 3-4:** Comparison of the baseline performance with the PSUB power subtraction algorithm. PSUB was only applied to the CRPZM.

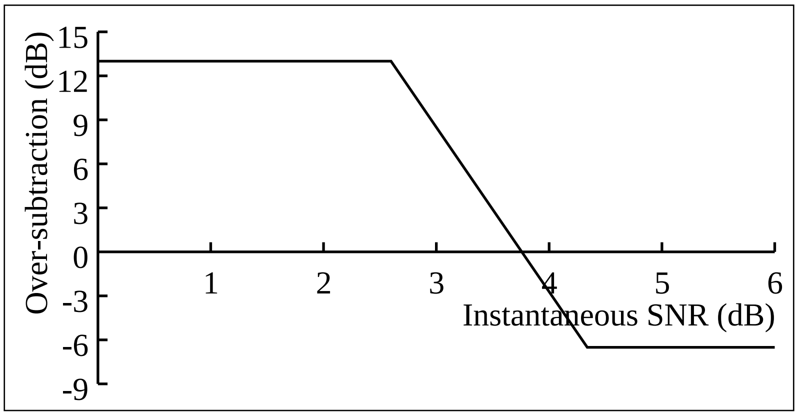



**Figure 3-7:** Amount of over and under subtraction used in the MSUB algorithm as a function of the instantaneous SNR.

$$\overline{\mathbf{Y}}(\omega_k) = \mathbf{Y}(\omega_k) - \mathbf{N}(\omega_k) \tag{3.12}$$

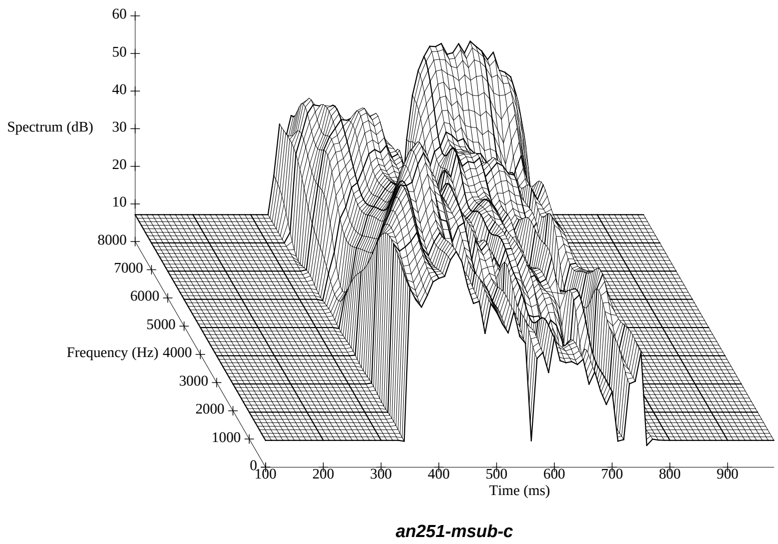
Then, the estimate of the log-energy of the restored signal in band $k$, $\hat{\mathbf{X}}(\omega_k)$, can be expressed as follows:

$$\hat{\mathbf{X}}(\omega_k) = \mathbf{Y}(\omega_k) + f(\overline{\mathbf{Y}}(\omega_k)) \tag{3.13}$$

Porter and Boll [67] used the assumption that the magnitude of the noisy speech had a Rayleigh distribution in order to derive his estimator. In our work, the transformation $f$ was derived directly from the stereo database as the transformation that minimized:

$$E\{[\mathbf{X}(\omega_k) - \hat{\mathbf{X}}(\omega_k)]^2\} = E\{[\mathbf{X}(\omega_k) - \mathbf{Y}(\omega_k) - f(\overline{\mathbf{Y}}(\omega_k))]^2\} \tag{3.14}$$

where the $\mathbf{Y}(\omega_k)$ are taken from the CRPZM speech and $\mathbf{X}(\omega_k)$ from the CLSTK speech for the same utterance. The function $f$ is the one that makes the CRPZM speech as similar to the CLSTK as possible in the sense of minimum mean squared error. To compute the new transformation curves, for every pair of stereo utterances from the database a gain normalization was performed so that the maximum zeroth cepstral component $g$ in both utterances is 0.



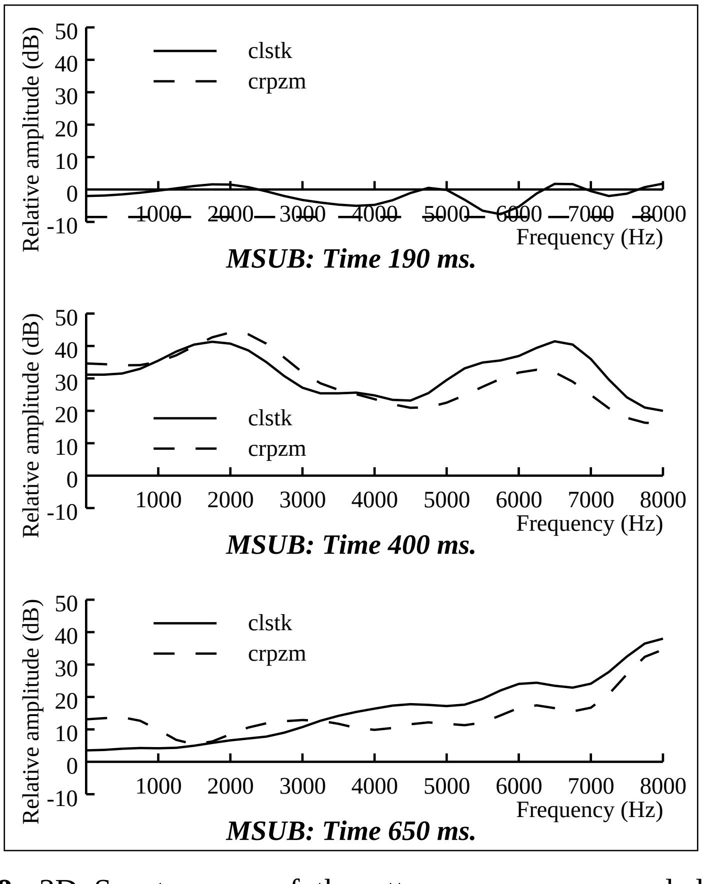




**Figure 3-8:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with MSUB algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **PSUB** | N/A | 38.6% | 70.6% | 70.1% |
| **MSUB** | N/A | 63.6% | 71.7% | 71.3% |

**Table 3-5:** Comparison of the baseline performance with the PSUB power subtraction and the MSUB magnitude subtraction algorithms. PSUB and MSUB were only applied to the CRPZM.

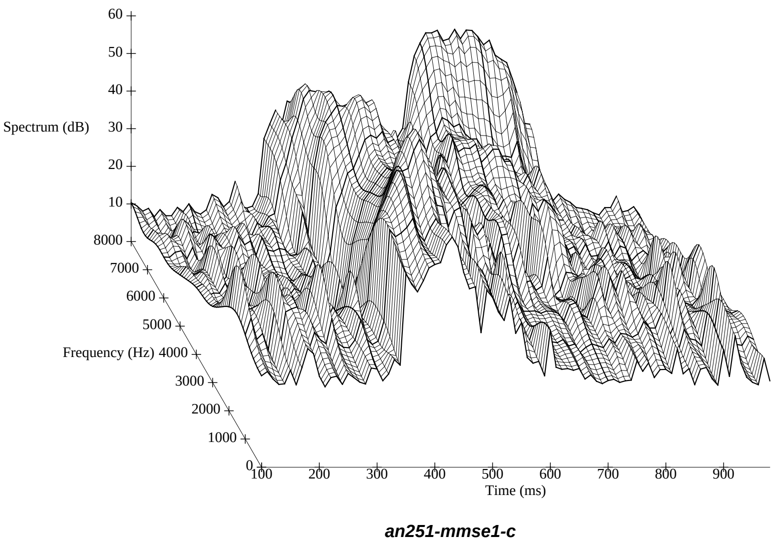
The expectation is replaced by a summation over all the frames in the training corpus for all 32 frequency bands. The function $f$ was discretized into 25 intervals separated 1 dB each. The result of the optimization was the value of $f$ for $j = 0, \ldots, 25$ in steps of $\Delta_{SNR} = 1dB$.

$$f[j] = \frac{\sum_{i=0}^{N-1} \sum_{k=0}^{k=32} [\mathbf{X}_i(\omega_k) - \mathbf{Y}_i(\omega_k)]\, \delta[\overline{\mathbf{Y}}_i(\omega_k) - j\Delta_{SNR}]}{\sum_{i=0}^{N-1} \sum_{k=0}^{k=32} \delta[\overline{\mathbf{Y}}_i(\omega_k) - j\Delta_{SNR}]} \tag{3.15}$$

where $\mathbf{X}_i(\omega_k)$ and $\mathbf{Y}_i(\omega_k)$ represent the log-energy of frame $i$ at frequency band $k$ for the CLSTK and CRPZM respectively. $\overline{\mathbf{Y}}_i(\omega_k)$ is the quantized SNR in frequency band $k$ of frame $i$ in the CRPZM, and $\delta[i]$ is the delta of Kronecker. The sum is carried out for all the $N$ frames in the database.

The MMSE1 algorithm applies the transformation to the 32 frequency bands of the CRPZM speech that minimizes the mean squared error relative to the CLSTK speech. The calculation of the compensating transformation was simpler with the use of the stereo database than with the equations derived by Porter and Boll [67] and Ephraim and Malah [14], without suffering from modeling inaccuracies.

In Figure 3-9 we can see the spectrum of the sample utterance for the CRPZM after processing by the MMSE1 algorithm. This algorithm performs noticeably better than the PSUB at low SNR (See frames at time 190 ms and 650 ms), although not as well as the MSUB algorithm. The evaluation results are shown in Table 3-6. The MMSE criterion outperforms the power subtraction rule as we expected. However, the MSUB algorithm still has higher accuracy despite not being goal-directed. We believe that the power of the MSUB algorithm stems from using different transformation curves for every frame SNR, as opposed to one single curve.



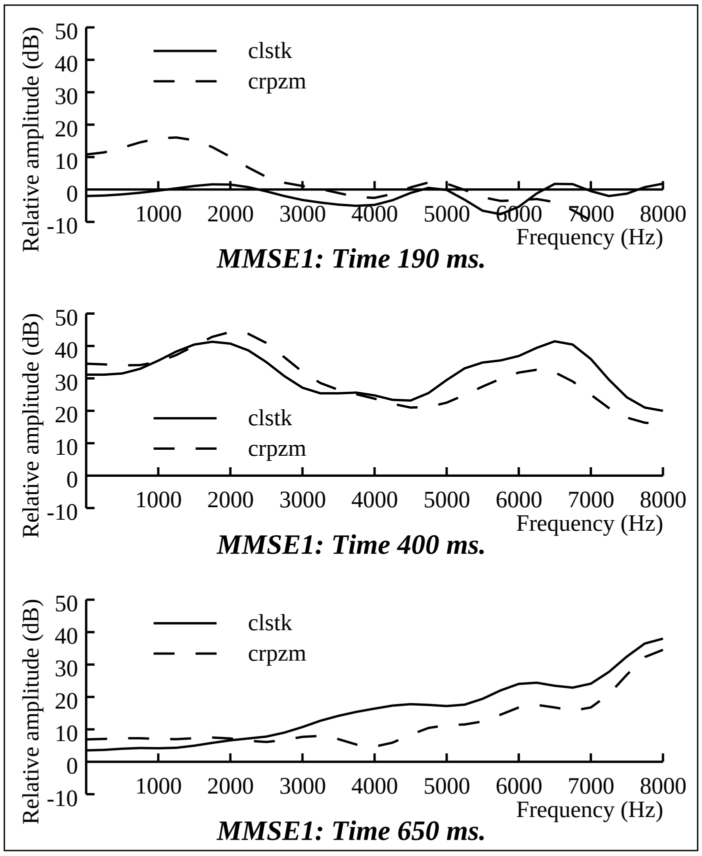




**Figure 3-9:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with cascade of EQUAL and MMSE1 algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **PSUB** | N/A | 38.6% | 70.6% | 70.1% |
| **MSUB** | N/A | 63.6% | 71.7% | 71.3% |
| **MMSE1** | N/A | 48.7% | 68.7% | 71.4% |

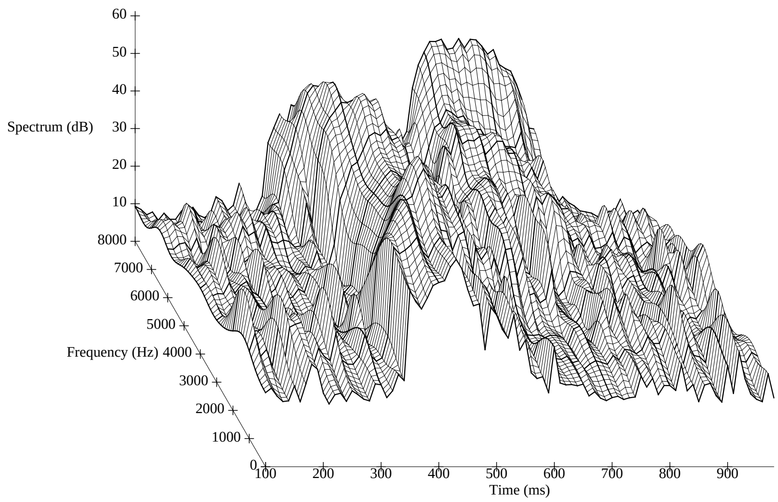
**Table 3-6:** Comparison of baseline performance with the PSUB, MSUB and MMSE1 algorithms. These algorithms were only applied to the CRPZM.

### 3.4.5. Cascade of EQUAL and MSUB

We applied a cascade of our channel equalization algorithm EQUAL and our noise suppression algorithms MMSE1 and MSUB, in an attempt to obtain an increased performance, since each algorithm combats a different kind of distortion.

Since equalization is a linear operation and the noise suppression algorithms are not, we opted to first perform the EQUAL algorithm and then the noise suppression. To apply the MMSE1 algorithm, another transformation function was calculated that minimized the squared error between the CLSTK speech and the equalized CRPZM speech. Figure 3-10 shows the sample utterance after this cascade of EQUAL and MMSE1. By comparing the frame at time 400 ms in Figures 3-10 and 3-9 we see that this processing is effective in combating spectral tilt too. The recognition results in Table 3-7 show the combined benefits of EQUAL and MMSE1. Unfortunately this combination does not perform better than the MSUB algorithm that only does noise suppression.

Following the success of the cascade of EQUAL and MMSE1, we proceeded to apply the cascade of EQUAL with our best noise subtraction algorithm MSUB. Again, we first performed the equalization and then the noise subtraction. In Figure 3-11 we present the sample utterance after this combined processing. The recognition results of this cascade of EQUAL and MSUB shown in Table 3-7 are disappointing, since the combined version is not doing better than the single MSUB. We believe that this is because of the non-linear interaction of noise and spectral tilt.




*an251-equalmmse1-c*

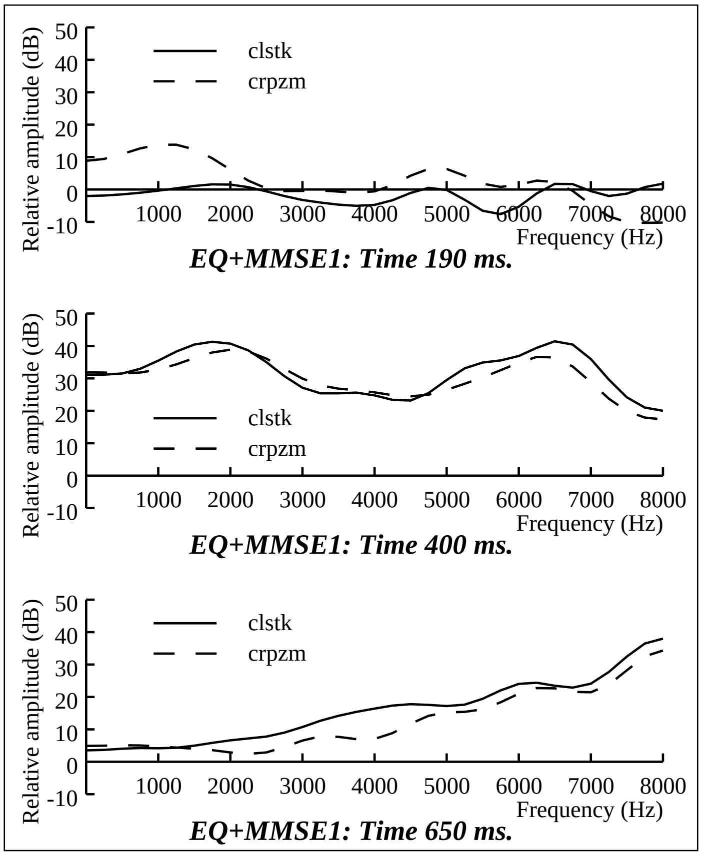
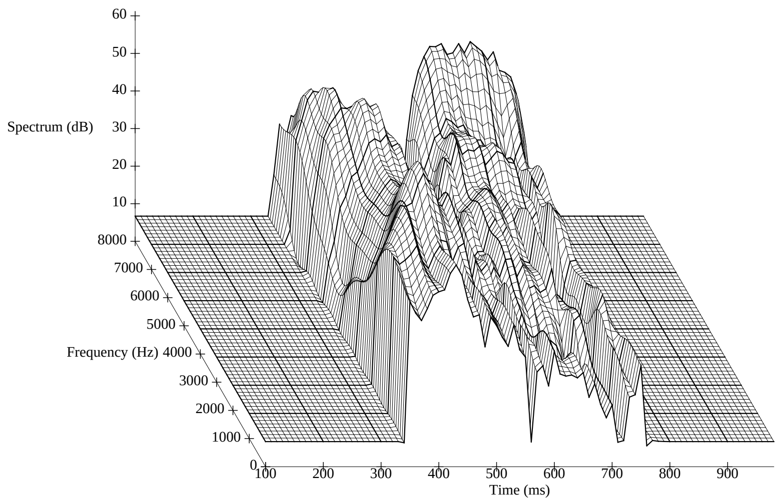



**Figure 3-10:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with the cascade of EQUAL and MMSE1 algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.




*an251-equalmsub-c*

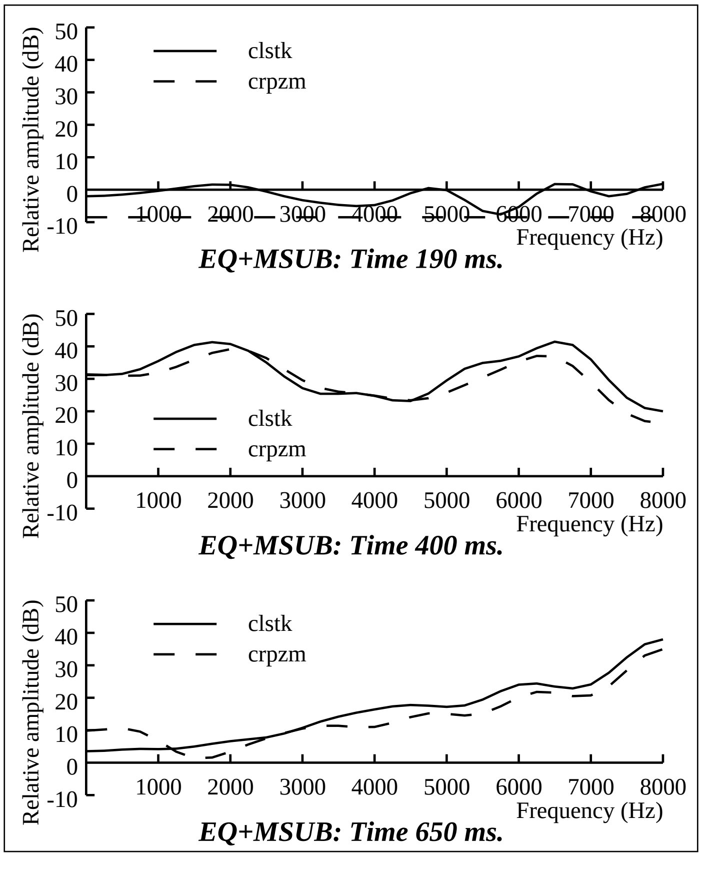



**Figure 3-11:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with the cascade of EQUAL and MSUB algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

### 3.4.6. Results and Discussion

The results of EQUAL, PSUB, MMSE1 and MSUB are shown in Table 3-7. There are several observations:

- The use of *a priori* information in MMSE1 resulted in a higher accuracy than the empirical power spectral subtraction rule in PSUB.
- The dependence on the instantaneous SNR was very beneficial, as noted by the difference in performance between PSUB and MSUB.
- The cascade of spectral equalization EQUAL and noise suppression MMSE1 performed considerably better than either of those alone. Unfortunately, this cascade did not yield any further improvement in recognition accuracy over the MSUB algorithm that only accomplishes noise suppression. Furthermore, a cascade of EQUAL and MSUB did not increase the accuracy over the MSUB algorithm alone. We conjecture that since noise is an additive distortion and channel equalization is a multiplicative distortion in the frequency domain, the two effects interact nonlinearly.

| **TRAIN<br>TEST** | **CLSTK<br>CLSTK** | **CLSTK<br>CRPZM** | **CRPZM<br>CLSTK** | **CRPZM<br>CRPZM** |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **EQUAL** | N/A | 38.3% | 50.9% | 76.5% |
| **PSUB** | N/A | 38.6% | 70.6% | 70.1% |
| **MSUB** | N/A | 63.6% | 71.7% | 71.3% |
| **MMSE1** | N/A | 48.7% | 68.7% | 71.4% |
| **EQ+MMSE1** | N/A | 61.4% | 75.8% | 74.3% |
| **EQ+MSUB** | N/A | 62.1% | 73.7% | 71.4% |

**Table 3-7:** Performance of different equalization and spectral subtraction algorithms. All the algorithms were applied only to the CRPZM.

As a way of unifying all the approaches, we can view the different subtraction algorithms as different transformation curves that relate the effective SNR of the input and output for every frequency channel. We note that the MSUB algorithm is not represented by a single curve by but a family of curves that depended on the instantaneous SNR of the frame. Some of these curves are shown in Figure 3-12.

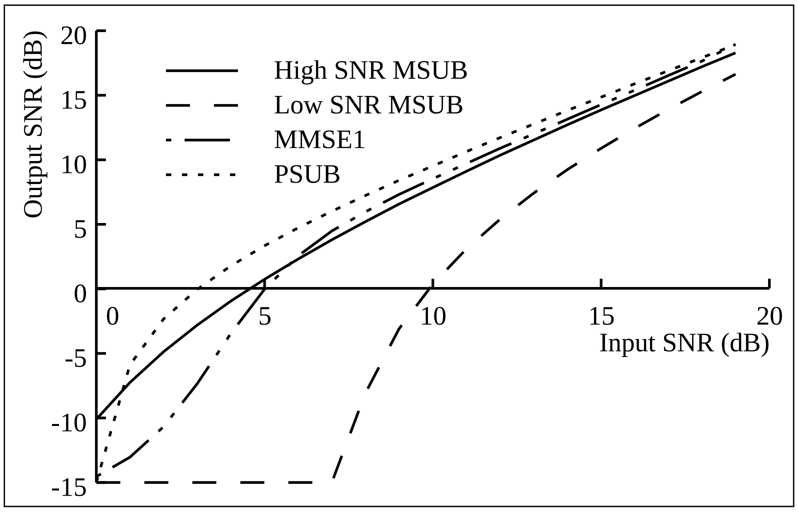



**Figure 3-12:** Input-Output transformation curves for PSUB, MSUB and MMSE1. The channel SNR is defined as the log-power of the signal in a frequency band minus the log-power of the noise in that band. The transformation for MSUB is not a single curve but a family of curves that depend on the total SNR for a given frame.

## 3.5. Summary

In this chapter we have presented several algorithms to increase the robustness of the system for the "cross conditions". Multi-Style training is a plausible alternative to the problem of robustness although at the expense of a lower recognition accuracy than when training and testing are done on the same environment.

The algorithm EQUAL applied a spectral equalization to the speech recorded with the CRPZM microphone to compensate for differences in transfer functions between training and test corpora.

A number of noise suppression techniques were discussed in this chapter. First, we described in detail the power spectral subtraction rule. An algorithm called PSUB directly implemented this rule in the logarithm domain. The MSUB algorithm was developed to combat the *musical noise* characteristic of standard spectral subtraction. Magnitude subtraction was used in MSUB as opposed to power subtraction because it provided a higher accuracy. The need for a uniform criterion in the front-end led us to try noise suppression techniques based on an MMSE criterion.

Cascading of equalization and spectral subtraction combined the improvements of both techniques although the recognition accuracy was still worse than for the MSUB algorithm. A cascade of EQUAL and MSUB did not perform better presumably because of the non-linear interaction of the equalization and noise subtraction.

For the most part these algorithms provide increasing degrees of compensation, but their recognition accuracy under the "cross" conditions is still worse than that obtained with the system is trained and tested on the CRPZM. The problem of the nonlinear interaction of the subtraction and normalization processes motivated us to consider new algorithms which jointly compensate for noise and filtering. We discuss several such algorithms in the next chapter.

# 4 Joint Compensation for Noise and Filtering

In this chapter we introduce the concept of joint normalization of noise and spectral tilt as well as the idea of preprocessing in the cepstral domain as opposed to the frequency domain. We first define the model of the environment that will be used in the remaining of the thesis and we introduce techniques that accomplish the processing in the cepstral domain.

We develop the MMSEN algorithm as an extension of the MMSE1 algorithm of Chapter 3. The MMSEN algorithm is able to cope with colored noise and spectral tilt jointly, and yields higher accuracy than a cascade of equalization and MMSE1.

By looking at the cepstral domain instead of the frequency domain we introduce the SDCN algorithm, that performs a cepstral compensation that depends on the SNR of the input frame exclusively. This algorithm is simple and effective.

## 4.1. A Model of the Environment

In this section we introduce the model of the environment as well as some notation conventions that will be used in the remaining of this dissertation. We will present relationships in the cepstral domain and define the concept of correction vectors.

The model of Figure 4-1 describes the two kinds of degradation that we have seen in the previous chapter: additive noise and linear filtering. We further assumed that the noise, which is colored[24] in general, was independent of the signal. Although we could have chosen to add the noise first and filter later, the model of Figure 4-1 introduces no

---

[24]*White* noise presents an even distribution of power for all frequencies, whereas *colored* noise has more power in some frequency bands than others.

loss of generality and has the advantage that observation of $y[m]$ when no signal is present provides us the noise directly and not filtered noise. For this work we will assume that the filter $h[m]$ is *time-invariant* and that the noise is *stationary* so that the parameters of the model are assumed to be fixed during the course of our observation.

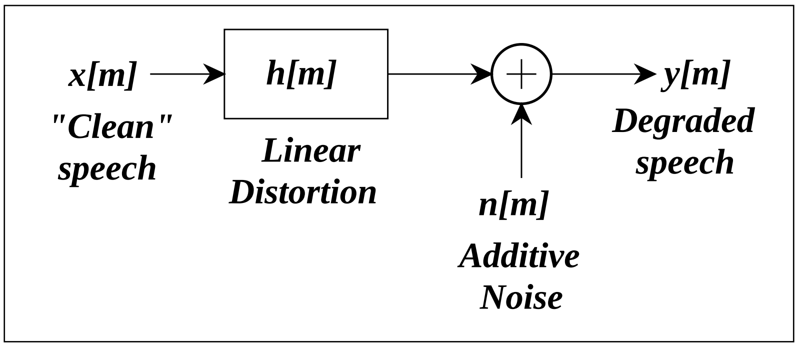



**Figure 4-1:** Model of the degradation.

If the input to the linear time-invariant filter $h[m]$ is a stationary random process $x[m]$ with Power Spectral Density (PSD) $X(\omega)$, we know [64] that the PSD of the output process $v[m]$ is

$$V(\omega) = X(\omega)\,|H(\omega)|^2 \tag{4.1}$$

Furthermore, if a stationary random process $n[i]$ that is uncorrelated with $v[i]$ is added to it, the PSD of the resulting random process $y[i]$ is

$$Y(\omega) = V(\omega) + N(\omega) \tag{4.2}$$

Combining (4.1) and (4.2) we obtain an expression that relates the PSD of the input, noise, output and the transfer function of the filter:

$$Y(\omega) = X(\omega)\,|H(\omega)|^2 + N(\omega) \tag{4.3}$$

Let's express Equation (4.3) in the cepstral domain rather than in the frequency domain. To do that, we need to define the following cepstrum vectors $\mathbf{x}$, $\mathbf{n}$, $\mathbf{y}$ and $\mathbf{q}$ as the following Inverse Discrete Fourier Transforms (IDFT):

$$\mathbf{x} = \text{IDFT}\,\{\ln X(\omega_k)\} \tag{4.4}$$
$$\mathbf{n} = \text{IDFT}\,\{\ln N(\omega_k)\} \tag{4.5}$$
$$\mathbf{y} = \text{IDFT}\,\{\ln Y(\omega_k)\} \tag{4.6}$$
$$\mathbf{q} = \text{IDFT}\,\{\ln |H(\omega_k)|^2\} \tag{4.7}$$

Taking natural logarithms and IDFT on (4.3) and after some algebraic manipulation using (4.4)-(4.7), we obtain:

$$\mathbf{y} = \mathbf{x} + \mathbf{q} + \mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{q}) \tag{4.8}$$

or alternatively

$$\mathbf{y} = \mathbf{n} + \mathbf{s}(\mathbf{x}, \mathbf{n}, \mathbf{q}) \tag{4.9}$$

where the correction vectors $\mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{q})$ and $\mathbf{s}(\mathbf{x}, \mathbf{n}, \mathbf{q})$ are given by

$$\mathbf{r}(\mathbf{x},\mathbf{n},\mathbf{q}) = \mathrm{IDFT}\ \{\ln\ (1 + e^{\mathrm{DFT}\ [\mathbf{n}-\mathbf{q}-\mathbf{x}]})\} \tag{4.10}$$

$$\mathbf{s}(\mathbf{x},\mathbf{n},\mathbf{q}) = \mathrm{IDFT}\ \{\ln\ (1 + e^{\mathrm{DFT}\ [\mathbf{x}+\mathbf{q}-\mathbf{n}]})\} \tag{4.11}$$

where (4.8) and (4.9) can be used interchangeably. The correction vectors **r** and **s** are uniquely related given the knowledge of **n**, **q** and **x**. In this work we will mainly use **r** and the expression (4.8).

Although Equations (4.8) and (4.9) involve only vector additions, the presence of the correction vectors of Equations (4.10) and (4.11) may at first indicate that the relationship was simpler in the frequency domain than in the cepstral domain. However, these vectors could be precomputed.

In summary, the model in Figure 4-1 can be characterized in the frequency domain by (4.3), or alternatively in the cepstral domain by (4.8) and (4.9) with the newly introduced correction vectors in (4.10) and (4.11).

## 4.2. Processing in the Frequency Domain: The MMSEN Algorithm

Although the approaches developed in Chapter 3 were relatively successful, they present some problems. The amount of noise subtraction given by the transformation curves of Figure 3-12 is the same for all frequencies, which is unable to deal with colored noise as in our case. To cope with these problems, we propose the MMSEN algorithm.

The MMSEN algorithm applies an *independent* transformation for every frequency band that minimizes the mean squared error between the CLSTK and the CRPZM speech. Equation (3.12) is still valid and Equation (3.13) becomes

$$\hat{\mathbf{X}}(\omega_k) = \mathbf{Y}(\omega_k) + f_k(\overline{\mathbf{Y}}(\omega_k)) \tag{4.12}$$

where there are 32 independent $f_k$ transformations that are chosen to minimize

$$E\{[\mathbf{X}(\omega_k) - \hat{\mathbf{X}}(\omega_k)]^2\} = E\{[\mathbf{X}(\omega_k) - \mathbf{Y}(\omega_k) - f_k(\overline{\mathbf{Y}}(\omega_k))]^2\} \tag{4.13}$$

Solution to Equation (4.13) yields

$$f_k[j] = \frac{\sum_{i=0}^{N-1} [\mathbf{S}_i(\omega_k) - \mathbf{X}_i(\omega_k)]\, \delta[\overline{\mathbf{Y}}_i(\omega_k) - j\Delta_{SNR}]}{\sum_{i=0}^{N-1} \delta[\overline{\mathbf{Y}}_i(\omega_k) - j\Delta_{SNR}]} \tag{4.14}$$

where $\mathbf{X}_i(\omega_k)$ and $\mathbf{Y}_i(\omega_k)$ represent the log-energy of frame $i$ at frequency band $k$ for the CLSTK and CRPZM respectively. $\overline{\mathbf{Y}}_i(\omega_k)$ is the SNR in frequency band $k$ of frame $i$ in the CRPZM. The sum is carried out for all the $N$ frames in the database.

To compute the new transformation curves, for every pair of stereo utterances from the database a gain normalization was performed so that the maximum $\mathbf{c}_{max}[0]$ in both utterances is 0. For every frequency band the difference between the CLSTK and the CRPZM was computed as a function of the channel SNR. Figure 4-2 shows the transformation curves for frequency bands of 0, 1, 2, 3, 4, 5, 6, and 8 kHz and its comparison with the MMSE curve averaged over all frequencies.

We can see that at low frequencies (below 4 kHz), the amount of noise subtraction is greater than at high frequencies (above 4 kHz) and thus the algorithm is able to handle the colored noise present in the CRPZM speech. As far as spectral tilt is concerned, low frequencies (below 4 kHz) are attenuated while high frequencies (above 4 kHz) are boosted and therefore channel equalization is achieved.

The results of this new MMSEN algorithm are presented in Table 4-1. Since only the CRPZM speech is modified, the most important figure for comparison purposes is the one obtained with CLSTK training and CRPZM testing. We believe that the MMSEN performs better than the EQUAL + MMSE1 algorithm because it deals better with the colored noise and it provides spectral equalization as well as noise subtraction in a *joint* fashion (Compare Figures 3-10 and 4-3, especially at time 190 ms). Actually, application of matched pairs tests, McNemar's test and analysis of variance by ranks (Gillick [27], Pallett *et al.* [63]) showed that the difference between 61.3% and 66.4% is statistically significant with a confidence level higher than 95%.

The MMSEN algorithm performs also moderately better than the MSUB algorithm, although the difference is not statistically significant. Both algorithms use a family of curves as opposed to a single transformation curve, the difference is that in the MMSEN algorithm the dependence is on the frequency band while in the MSUB algorithm the dependence is on the input SNR. Conceivably a greater improvement could be attained if both dependencies are used together.

The same example that was presented in the previous chapter is shown in Figure 4-3 when processed by the MMSEN algorithm. The resulting noise is now approximately white, unlike the residual noise in the PSUB and MMSE1 algorithms, because of the different transformation used for every frequency band.

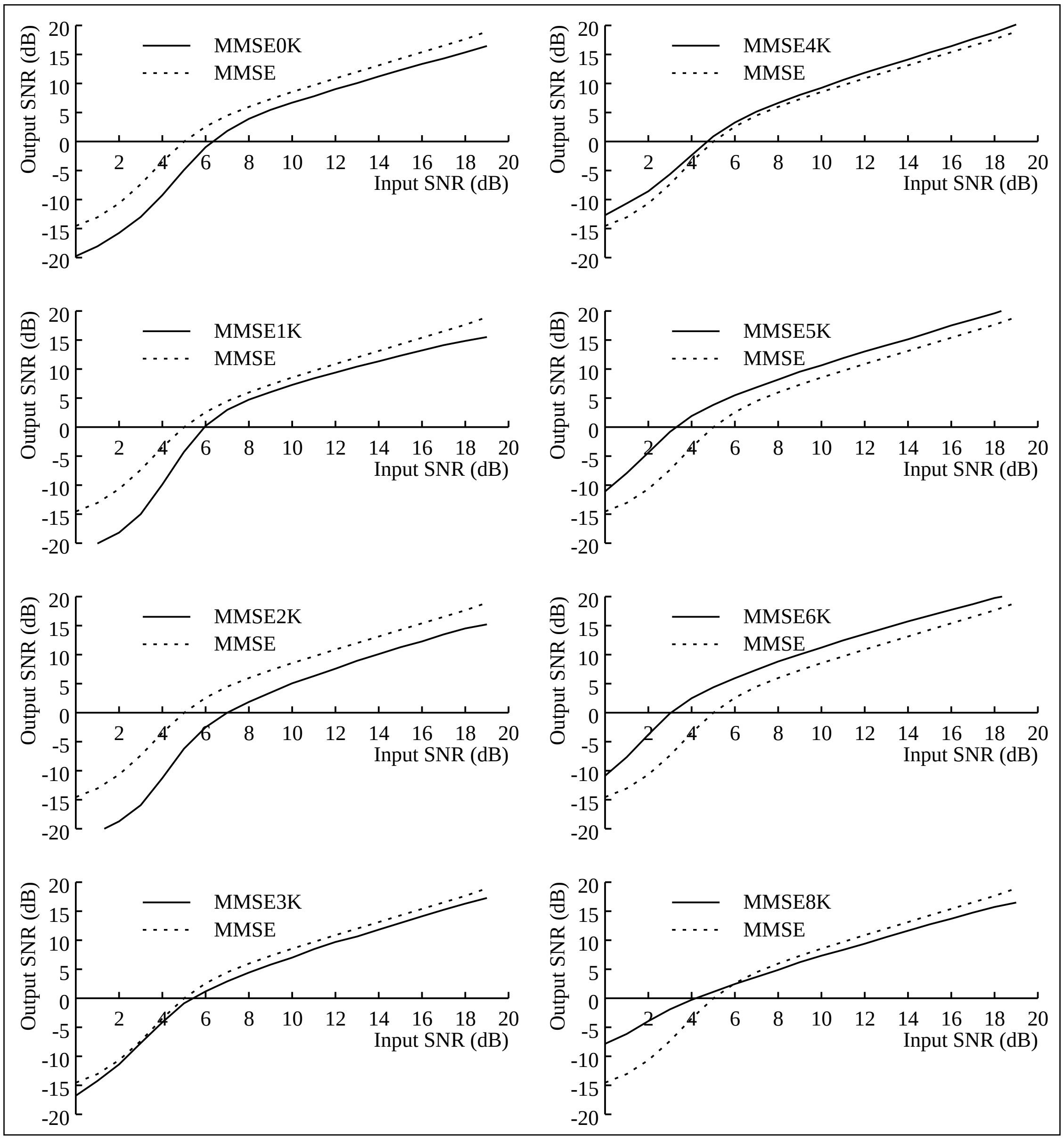



**Figure 4-2:** Comparison between the transformation curve MMSE fixed for all frequencies and the corresponding transformations for different frequencies: 0, 1, 2, 3, 4, 5, 6 and 8 kHz. The curves give the input-output relation between the SNR at a frequency band. It can be seen that more noise subtraction is done at low frequencies than at high frequencies. Also, low frequencies are attenuated more at high SNR, to compensate for spectral tilt.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **EQ+MMSE1** | N/A | 61.4% | 75.8% | 74.3% |
| **MSUB** | N/A | 63.6% | 71.7% | 71.3% |
| **MMSEN** | N/A | 66.4% | 75.5% | 72.3% |

**Table 4-1:** Performance of the MMSEN compared with the Baseline and the MMSE1 and MSUB algorithms.

## 4.3. Processing in the Cepstral Domain: The SDCN Algorithm

The MMSEN algorithm performs *joint* noise subtraction and spectral equalization, the processing has to be done in the frequency domain for a number of bands (32 in our case). Many other systems that obtain mel-scale filter bank outputs typically have between 20 and 40 channels. Since SPHINX uses the LPC-cepstrum, the environmental normalization requires two DFTs plus 32 table look-ups. The question that arises is whether it is possible to perform this normalization directly in the cepstral domain.

The *SNR-Dependent Cepstral Normalization* algorithm, SDCN, is an algorithm that operates directly in the cepstral domain by adding a compensation vector that depends exclusively on the SNR of the input frame. A justification for this idea within our model of the environment is given now.

Let **z** be a noisy estimate of **y** as defined in (4.8) obtained through our spectral estimation algorithm:

$$\mathbf{z} = \mathbf{y} + \mathbf{e} \tag{4.15}$$

where **e**, the estimation error, is a random vector. Our goal is to recover the uncorrupted vector **x** of an utterance given the observation **z** and our knowledge of the environment **n** and **q**.

$$\mathbf{x} = \mathbf{z} - \mathbf{e} - \mathbf{q} - \mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{q}) \tag{4.16}$$

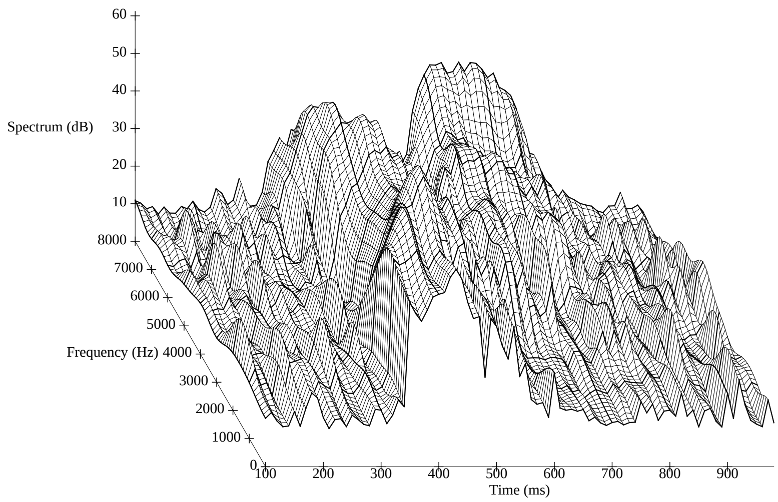
Unfortunately, given the non-linear expression (4.10), it is not possible to obtain a close-form solution for **x**. One possible approximation would be to assume that the correction vector

$$\mathbf{w} = -\mathbf{e} - \mathbf{q} - \mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{q}) \tag{4.17}$$

depends exclusively on the instantaneous SNR of the input frame:

$$\hat{\mathbf{x}} = \mathbf{z} + \mathbf{w}(SNR) \tag{4.18}$$

This is the basic equation for the SDCN algorithm.




***an251-mmsen-c***

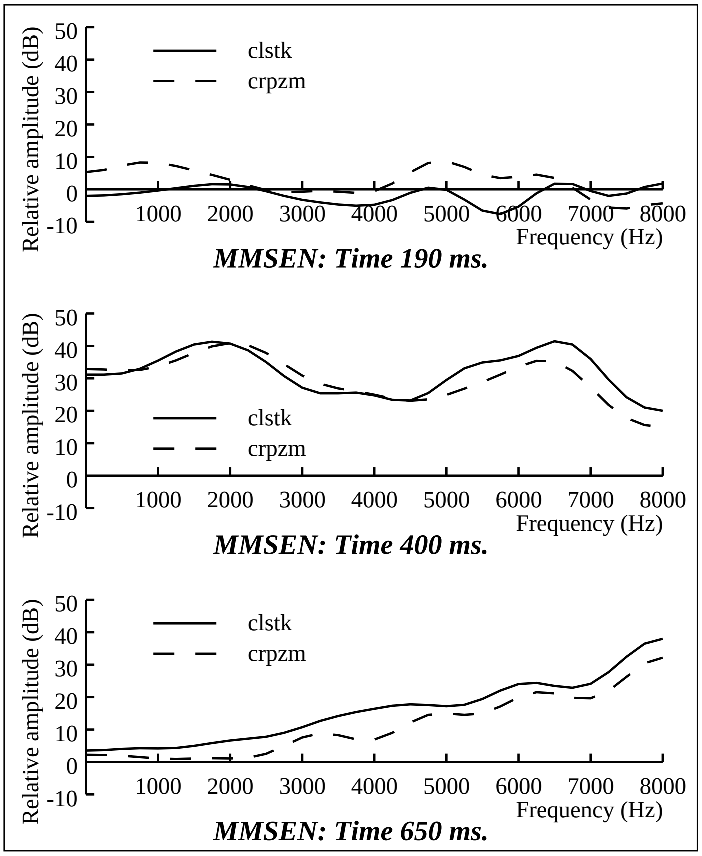



**Figure 4-3:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with MMSEN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

Let's consider a physical interpretation of the algorithm. Although inspection of Equation (4.18) may lead to think that a crude approximation of Equation (4.10) has been made, we will show that the SDCN algorithm is asymptotically correct. At high SNR, inspection of Equations (4.3), (4.8) and (4.10) indicates that $\mathbf{x}[0] + \mathbf{q}[0] \gg \mathbf{n}[0]$, $\mathbf{r} \approx \mathbf{0}$, and $\mathbf{y} \approx \mathbf{x} + \mathbf{q}$. On the other hand at low SNR, $\mathbf{x}[0] + \mathbf{q}[0] \ll \mathbf{n}[0]$ and $\mathbf{y} \approx \mathbf{n}$. Hence, the SDCN algorithm performs spectral equalization at high SNR and noise suppression at low SNR. However, it is evident that at intermediate SNR, the SDCN algorithm can only be an approximation. Another simplification that we undertake is to estimate the input SNR as $\mathbf{z}[0] - \mathbf{n}[0]$. Although this is not the true instantaneous signal-to-noise ratio, it is related to it and easier to compute in our case.

The problem now is how to estimate the compensation vectors $\mathbf{w}(SNR)$. The goal is to transform the CRPZM speech so that it looks like the CLSTK speech. Therefore according to this criterion, the correction vectors were estimated by computing the average difference between cepstral vectors for the CRPZM speech versus the cepstral vectors for the CLSTK speech on a frame-by-frame basis as a function of the input SNR.

$$\mathbf{w}[j,k] = \frac{\sum_{i=0}^{N-1} (\mathbf{x}_i[j] - \mathbf{z}_i[j])\, \delta[SNR_i - k\Delta_{SNR}]}{\sum_{i=0}^{N-1} \delta[SNR_i - k\Delta_{SNR}]} \tag{4.19}$$

where $\mathbf{s}_i$ and $\mathbf{x}_i$ represent the cepstrum vectors at frame $i$ for the CLSTK and CRPZM respectively. $SNR_i$ is the SNR of frame $i$ in the CRPZM. The sum is carried out for all the $N$ frames in the database.

In the general case the correction vectors can be estimated by computing the difference between the cepstral vectors of the test environment and the ones of a standard acoustical environment from simultaneous stereo recordings. The resulting correction vectors for our census database are shown in Figure 4-4.

Figure 4-5 shows the same example that was presented in the previous chapter when processed by the SDCN algorithm. We note that frames with low SNR (Compare frames at time 190 ms and 650 ms for Figures 4-3 and 4-5) are smoother in this case than in the MMSEN case.

The performance of the SDCN algorithm is shown in Table 4-2, together with the baseline condition and the previously described MMSEN. This algorithm offers the highest accuracy of all the ones described so far. However, tests of statistical significance showed that the difference between 66.4% and 67.2% may be due to chance.

As can be seen in Figure 4-4, the high-order components of $\mathbf{w}$ are small in magnitude and don't vary substantially with the SNR. Inspired by this fact, an experiment was carried out with the SDCN algorithm in which only the first $p$ components of $\mathbf{w}$ were

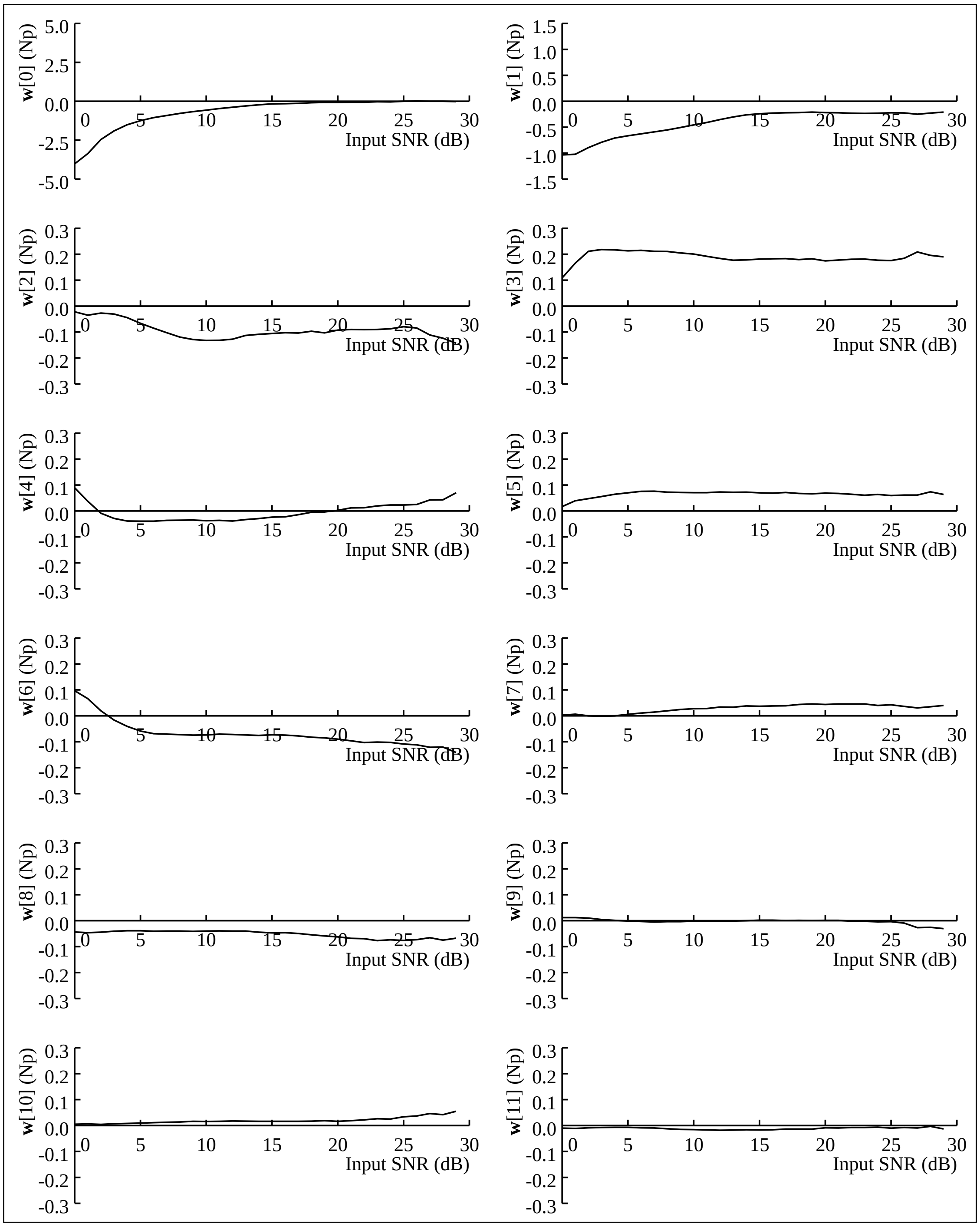
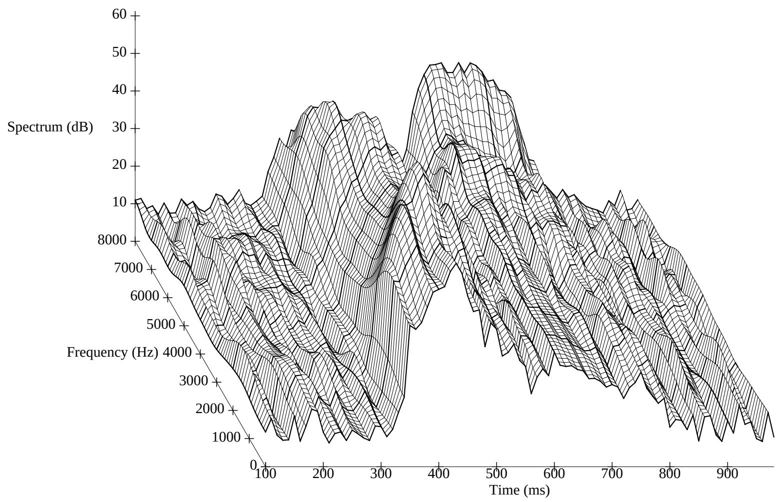



**Figure 4-4:** Correction vector **w**[*i*] in Np as a function of the instantaneous SNR in dB. Note the different scale used for **w**[0] and **w**[1], as they are the correction vectors varying the most. Correction vector **w**[12] is not shown.

used for the compensation. The results are shown in Figure 4-6 for the case of training on the CLSTK and testing on the CRPZM. It is quite remarkable that only **w**[0] and **w**[1] are




*an251-sdcn-c*

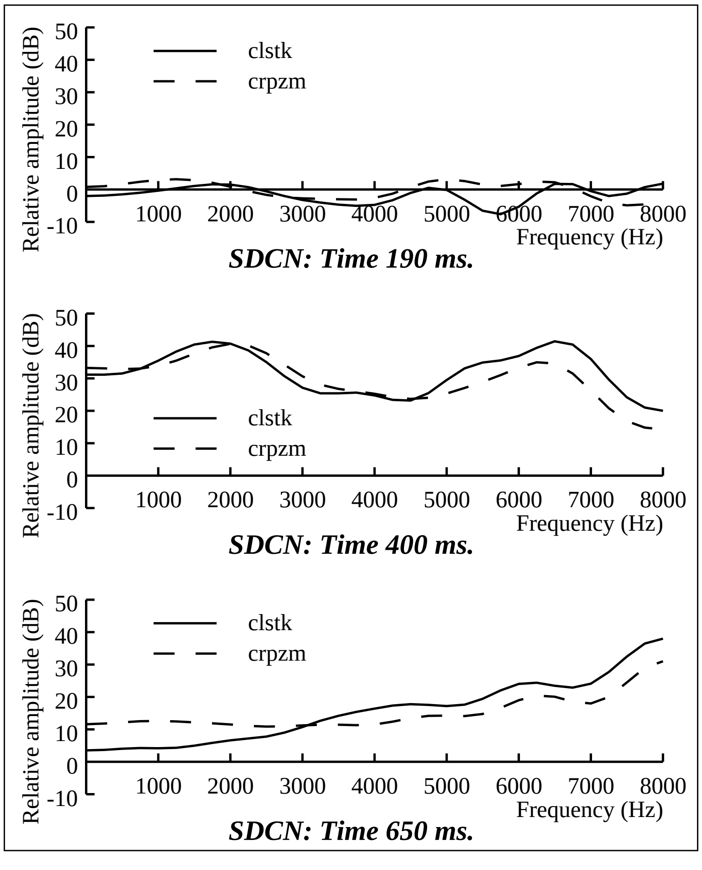



**Figure 4-5:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with SDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **MMSEN** | N/A | 66.4% | 75.5% | 72.3% |
| **SDCN** | N/A | 67.2% | 76.4% | 75.5% |

**Table 4-2:** Performance of the MMSEN and SDCN algorithms when compared with the baseline.

necessary. Use of the remaining components yields no further improvement in word accuracy.

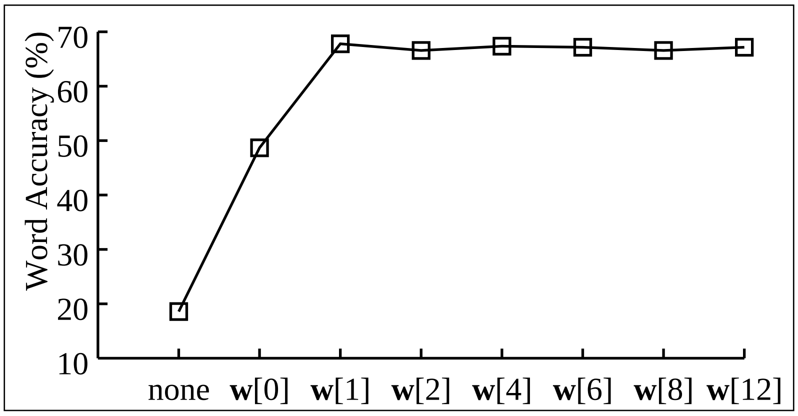



**Figure 4-6:** Word accuracy of the SDCN algorithm as a function of the number of correction vectors used when trained on the CLSTK and tested on the CRPZM. None means that no correction is applied (baseline), **w**[0] means that only **c**[0] was compensated, **w**[1] that both **c**[0] and **c**[1] were compensated, etc.

## 4.4. Summary

In this chapter we have addressed the problem of *joint* compensation for noise and spectral tilt. The two algorithms presented, MMSEN and SDCN, successfully achieve the compensations in the frequency domain and the cepstral domain respectively. Both algorithms exhibit a higher accuracy than the algorithms that perform independent compensation for noise and spectral tilt described in Chapter 3.

There are some similarities between the two algorithms. They both make use of several transformation curves, one for each dimension in the space. The dependency on the MMSEN algorithm is on the channel SNR while for the SDCN algorithm it is on the frame SNR[25]. Typically, approaches that work in the frequency domain have a larger dimension (N = 32 in our case) than the ones that operate in the cepstral domain (N = 12 in our case).

The curves for the MMSEN algorithm could be viewed as transformation curves as in Figure 4-2, or alternatively as an additive correction on the input value. Given the cepstral structure, SDCN can be viewed as strictly an additive correction ($\mathbf{w}[i]$ does not depend on $\mathbf{c}[i]$ but on the frame SNR).

Finally, both SDCN and MMSEN seem to perform equally well, but SDCN requires only 2 corrections whereas MMSEN needs 32.

Even though the performance of the SDCN algorithm is quite acceptable, it requires a stereo database of our standard environment and the new environment to train the correction vectors. Since in a real situation such a database may not be available, the SDCN algorithm is not *environment-independent*.

We have also found that the above techniques produce many output frames that do not constitute legitimate speech vectors, especially at low SNR, because they do not take into account correlations across frequency bands or cepstral components. In the next chapter we discuss algorithms that explicitly consider the spectral profile of the compensated speech and that do not need a stereo database.

---

[25] The instantaneous channel SNR is defined as the difference between the log-energy of the signal and the log-energy of the noise for a frequency band at a reference time. The frame SNR is defined as the difference between the $\mathbf{c}[0]$ of the input frame and the $\mathbf{c}[0]$ of the noise at a reference time.

# 5 CDCN Algorithm

Although the SDCN technique performs acceptably, it has the disadvantage that new microphones must be "calibrated" by collecting long-term statistics from a new stereo database. Since this stereo database will not be available in general, SDCN cannot adapt to a new environment. A new algorithm, *Codeword-Dependent Cepstral Normalization* (CDCN), was proposed to circumvent these problems, and will be the topic of this chapter.

In CDCN, the noise and channel equalization vectors are estimated so as to best match the acoustic space of the input speech frames with the acoustic space obtained by transforming a *universal* space with the environmental parameters, as depicted in Figure 5-1. This universal acoustic space is defined as the distribution of speech frames under a normalized clean environment, and is represented by a codebook of cepstrum vectors. A maximum likelihood criterion is used to estimate the environmental parameters that transform the standard acoustical ambiance into the acoustic space of the current environment.

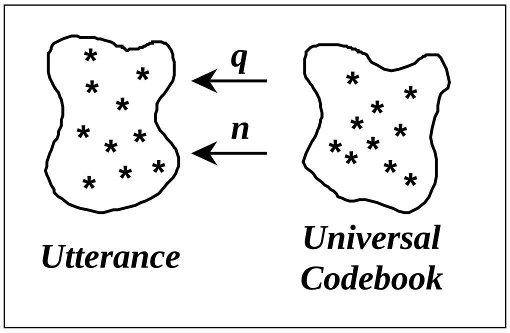



**Figure 5-1:** CDCN estimates the noise **n** and channel equalization **q** that best transform the universal codebook into the ensemble of input frames of the current utterance.

Although the MMSEN and the SDCN algorithms were able to *jointly* normalize for noise and spectral tilt, the corrections are the same for all speech vectors. The MMSEN algorithm assumed that different frequency bands were uncorrelated, and that sometimes produced restored vectors that were not legitimate speech vectors, especially at low SNR. Likewise, the SDCN algorithm assumes that different cepstral components are uncorrelated, and it produced restored cepstral vectors that were not legitimate speech cepstrum vectors, again especially at low SNR.

This use of *a priori* information for the speech vector was simultaneously proposed by Erell and Weintraub [18] and Acero and Stern [1]. Erell and Weintraub modeled the vector in the spectral domain as a mixture of Gaussians while Acero and Stern used a mixture of Gaussians in the cepstral domain. These models remove the assumption of independence across frequency bands (like the MMSEN) or across cepstrum coefficients (like the SDCN).

These two approaches are similar, one working in the spectral domain and the other in the cepstral domain. Both of them clustered the acoustic space so that every cluster was transformed differently. In both cases an MMSE criterion was used for the restored vector that was essentially a weighted sum of the vectors transformed by all the different cluster transformations. One major difference is that in Erell's work only the noise suppression was performed with the above criterion while the spectral equalization was done separately, while in the CDCN algorithm both are done *jointly*. The other difference is that in Erell's work the estimation of the environment and especially the channel equalization, is done by taking long-term averages, so it is not able to *adapt* to a new environment as the CDCN algorithm does. However, there is no fundamental reason why processing in the frequency domain could not perform joint compensation and adapt to new environments.

First we define the framework we will be using and an outline of the CDCN algorithm with its two steps: MMSE estimator for the cepstrum and ML[26] estimation of the environmental parameters. Then we present a summary of the algorithm and evaluate the CDCN algorithm.

[26]ML, Maximum Likelihood of the desired estimate, is a procedure used in estimation theory that is based on the maximization of the probability density function with respect to a parameter.

## 5.1. Introduction to the CDCN Algorithm

In this section we present the framework of the CDCN algorithm as well as the approximations that will be used. We start with a general MAP approach and then justify the decoupling of the estimation of the environmental parameters from the string decoding process. We also discuss the effect of using discrete HMMs as SPHINX does, rather than semi-continuous HMMs, especially at low SNRs.

The objective of the speech recognition system is to find the most likely string $S$ and environmental parameters $\mathbf{n}$ and $\mathbf{q}$, from an ensemble of $N$ cepstrum vectors $Z = \mathbf{z}_0,...,\mathbf{z}_{N-1}$. We will assume that during the observation period, both $\mathbf{n}$ and $\mathbf{q}$ are constant, which is to say that the noise is stationary in that segment of time and that the filter is not varying either, so that $S$, $\mathbf{n}$ and $\mathbf{q}$ are chosen to maximize

$$p(S,\mathbf{n},\mathbf{q}|Z) = \frac{p(S,Z|\mathbf{n},\mathbf{q})\,p(\mathbf{n},\mathbf{q})}{p(Z)} \tag{5.1}$$

which is equivalent to maximizing

$$p(S,Z|\mathbf{n},\mathbf{q}) \tag{5.2}$$

if no *a priori* knowledge of $\mathbf{n}$ and $\mathbf{q}$ is available. By assuming independence across frames $p(S,Z|\mathbf{n},\mathbf{q})$ can be expressed as

$$p(S,Z|\mathbf{n},\mathbf{q}) = p(S|\mathbf{n},\mathbf{q})\,p(Z|S,\mathbf{n},\mathbf{q}) = p(S)\prod_{i=0}^{N-1} p(\mathbf{z}_i|S,\mathbf{n},\mathbf{q}) \tag{5.3}$$

since $p(S|\mathbf{n},\mathbf{q})$ is not a function of $\mathbf{n}$ and $\mathbf{q}$.

One of the reasons the spectral subtraction schemes did not perform well is that some of the processed frames were not legitimate speech vectors, especially at low SNR. This occurred because no use was made of *a priori* information about the speech. A widely used assumption for the *pdf* of the cepstrum $\mathbf{x}$ is a mixture of $K$ Gaussian densities with means $\mathbf{c}[k]$, covariance matrices $\Sigma_k$, and weights $P[k]$:

$$p(\mathbf{x}) = \sum_{k=0}^{K-1} P[k]\, p(\mathbf{x}|k) = \sum_{k=0}^{K-1} P[k]\, N_{\mathbf{x}}(\mathbf{c}[k],\Sigma_k) \tag{5.4}$$

Under these model assumptions $p(\mathbf{z}_i|S,\mathbf{n},\mathbf{q})$ is given by

$$\begin{aligned} p(\mathbf{z}_i|S,\mathbf{n},\mathbf{q}) &= \sum_{k=0}^{K-1} \int p(\mathbf{z}_i,\mathbf{x}_i,k|S,\mathbf{n},\mathbf{q})\,d\mathbf{x}_i \\ &= \sum_{k=0}^{K-1} \int p(\mathbf{z}_i|\mathbf{x}_i,k,S,\mathbf{n},\mathbf{q})\,p(\mathbf{x}_i,k|S,\mathbf{n},\mathbf{q})\,d\mathbf{x}_i \\ &= \sum_{k=0}^{K-1} \int p(\mathbf{z}_i|\mathbf{x}_i,k,\mathbf{n},\mathbf{q})\,p(\mathbf{x}_i|k)P_i[k|S]\,d\mathbf{x}_i \end{aligned} \tag{5.5}$$

since $p(\mathbf{z}_i|\mathbf{x}_i,k,S,\mathbf{n},\mathbf{q})$ is not a function of the string S and $p(\mathbf{x}_i|k,S,\mathbf{n},\mathbf{q})$ does not depend on $\mathbf{n}$ and $\mathbf{q}$ either.

For a vector $\mathbf{x}_i$ coming from mixture $k$ given $\mathbf{n}$ and $\mathbf{q}$, we can obtain $\mathbf{y}_i$ as defined by Equation (4.8). In Appendix D we model $\mathbf{z}_i$ as a Gaussian random vector $N_{z_i}(\mathbf{y}_i, \Gamma)$, due to a finite data sample in our spectral estimation. In Appendix E.2 we show that the integral in (5.5) can be approximated by

$$\int p(\mathbf{z}_i | \mathbf{x}_i, k, \mathbf{n}, \mathbf{q}) p(\mathbf{x}_i | k) d\mathbf{x}_i = N_{\mathbf{z}_i}(\mathbf{q} + \mathbf{r}[k] + \mathbf{c}[k], \Gamma + \Sigma_k) \tag{5.6}$$

by assuming that the correction vector $\mathbf{r}[k]$ is constant. So Equation (5.3) can be expressed as

$$p(S, Z | \mathbf{n}, \mathbf{q}) = p(S) \prod_{i=0}^{N-1} \sum_{k=0}^{K-1} N_{\mathbf{z}_i}(\mathbf{q} + \mathbf{r}[k] + \mathbf{c}[k], \Gamma + \Sigma_k) P_i[k | S] \tag{5.7}$$

So the problem of the recognition of a sequence of corrupted vectors can be solved by integrating the acoustic model $p_{\mathbf{z}_i}(\mathbf{z}_i) = N_{\mathbf{z}_i}(\mathbf{q} + \mathbf{r}[k] + \mathbf{c}[k], \Gamma + \Sigma_k)$, the HMM model $P_i[k | S]$, and the grammar information $P(S)$ in a semi-continuous HMM framework.

### 5.1.1. Using Only Acoustic Information

Finding the ML estimates of $\mathbf{n}$ and $\mathbf{q}$ in Equation (5.7) requires a complicated search involving the HMM models and the grammar, that is, a joint optimization of the string $S$ and $\mathbf{n}$ and $\mathbf{q}$. We aim to solve the maximization without resorting to the HMM models, just considering the acoustic information. Although this will not be as optimal, we opted for this solution for expediency. We define

$$P_i[k] = E_S\{ p(S) P_i[k | S]\} \tag{5.8}$$

as the average mixture probabilities. In the absence of other information, these average probabilities will be the same for all $i$.

Unfortunately, finding the ML estimate of $\mathbf{n}$ in Equation (5.7) seems difficult since $\mathbf{n}$ is only implicitly present in the correction vectors $\mathbf{r}$. To make the presence of $\mathbf{n}$ explicit we will consider two different *phones* when taking the expectation in Equation (5.8): the noise model $\nu$ and the speech model $\xi$ (any other *phone*). With this assumption

$$p(\mathbf{z}_i | \mathbf{n}, \mathbf{q}) = E_S[p(S, \mathbf{z}_i | \mathbf{n}, \mathbf{q})] = p(\nu) p(\mathbf{z}_i | \mathbf{n}, \mathbf{q}, \nu) + p(\xi) p(\mathbf{z}_i | \mathbf{n}, \mathbf{q}, \xi)$$

$$= p(\nu) p_i(0 | \nu) N_{\mathbf{z}_i}(\mathbf{n} + \mathbf{s}[0], \Gamma) + p(\xi) \sum_{k=1}^{K-1} p_i(k | \xi) N_{\mathbf{z}_i}(\mathbf{q} + \mathbf{r}[k] + \mathbf{c}[k], \Gamma + \Sigma_k) \tag{5.9}$$

where we have assumed that the HMM model for the noise is so sharp that contains only one mixture. For that case we have used the alternate expression in terms of $\mathbf{n}$ and $\mathbf{s}$. If the noise is stationary within an utterance, one mixture is sufficient to represent it, so that this is totally general. We have also set $\Sigma_0$ to 0 by assuming that the variability observed in the noise samples comes from the spectral estimation process exclusively.

### 5.1.2. Using Discrete Models

However, since SPHINX uses discrete HMM models, replacing the summation in (5.7) by only one term will lead to errors, especially at low SNRs. For high SNR the correction vectors are approximately zero and the mixtures $N_{\mathbf{z}_i}(\mathbf{q} + \mathbf{r}[k] + \mathbf{c}[k], \Gamma + \Sigma_k)$ are well separated, so that substituting the sum with the maximum is a good approximation. Unfortunately, when the SNR is low, the correction vectors are significant and many codewords in the original space are transformed into essentially the same codeword, so that by picking the maximum we are actually underestimating the sum.

To solve this problem we opted to choose the MMSE estimate, so that a vector $\mathbf{x}_i$ is obtained for every $\mathbf{z}_i$ as that which minimizes the squared error. Our experiments confirmed that doing this is considerably better than picking the maximum in (5.7).

## 5.2. MMSE Estimator of the Cepstral Vector

In this Section, we will derive an MMSE estimator for the restored cepstral vector $\hat{\mathbf{x}}$ by using the mixture model. In this case we will assume that the environmental parameters $\mathbf{n}$ and $\mathbf{q}$ are known.

In Appendix E.1 it is shown that the probability density function $p(\mathbf{x}|\mathbf{z}, \mathbf{n}, \mathbf{q})$ has the form:

$$p(\mathbf{x}|\mathbf{z}, \mathbf{n}, \mathbf{q}) = \frac{\sum_{k=0}^{K-1} P[k]\, p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)}{\sum_{k=0}^{K-1} P[k] \int p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)\, d\mathbf{x}} \tag{5.10}$$

This *a posteriori* probability gives us all the information needed for a classification problem. A MAP estimator would imply in this case maximizing a sum of Gaussians. We will devote ourselves to obtaining an MMSE estimate for $\mathbf{x}$. From (5.10) it is easy to obtain the MMSE estimate as:

$$\hat{\mathbf{x}}_{MMSE} = E[\mathbf{x}\,|\mathbf{z}, \mathbf{n}, \mathbf{q}] = \frac{\sum_{k=0}^{K-1} P[k] \int \mathbf{x}\, p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)\, d\mathbf{x}}{\sum_{k=0}^{K-1} P[k] \int p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)\, d\mathbf{x}} \tag{5.11}$$

where $p(\mathbf{x}|k)$ is the *pdf* of the $k^{th}$ mixture and $p(\mathbf{z}|\mathbf{x}, k, \mathbf{n}, \mathbf{q})$ is the *pdf* of the spectral estimator $p(\mathbf{z}|\mathbf{y})$, both of which are assumed to be Gaussian random vectors.

Under the assumption that $p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)$ and $p(\mathbf{x}|k)$ are Gaussian densities, the product of the two can be expressed as follows (See Appendix E.2):

$$p(\mathbf{z}\,|\,\mathbf{x},\mathbf{n},\mathbf{q},k)\;p(\mathbf{x}\,|\,k) = N_{\mathbf{x}}(\mathbf{b}(\mathbf{x},k),(\Gamma^{-1}+\Sigma_k^{-1})^{-1})\;N_{\mathbf{z}}(\mathbf{q}+\mathbf{r}(\mathbf{x})+\mathbf{c}[k],\Gamma+\Sigma_k) \tag{5.12}$$

where the vector $\mathbf{b}(\mathbf{x},k)$ is given by

$$\mathbf{b}(\mathbf{x},k) = \Sigma_k(\Sigma_k+\Gamma)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})) + \Gamma(\Sigma_k+\Gamma)^{-1}\mathbf{c}[k] \tag{5.13}$$

and the quantity $d(\mathbf{x},k)$ is defined as

$$d(\mathbf{x},k) = (\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})-\mathbf{c}[k])^T(\Gamma+\Sigma_k)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})-\mathbf{c}[k]) \tag{5.14}$$

Unfortunately, it is not possible to obtain a closed-form expression for the integrals in (5.11) even with (5.12) because $\mathbf{r}$ depends on $\mathbf{x}$ as given by (4.10), and some approximations have to be made. We will assume that the correction vector $\mathbf{r}$ is constant for all vectors in a mixture. The finer the partition of the space is, the better the approximation. For the mixture $k$, the correction vector is assumed to be

$$\mathbf{r}[k] = \mathbf{r}(\mathbf{c}[k]) \tag{5.15}$$

where the mean of the mixture is taken to compute the correction vector from (4.10). Equations (5.13) and (5.14) are transformed according to (5.15):

$$\mathbf{b}[k] = \Sigma_k(\Sigma_k+\Gamma)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}[k]) + \Gamma(\Sigma_k+\Gamma)^{-1}\mathbf{c}[k] \tag{5.16}$$

$$d[k] = (\mathbf{z}-\mathbf{q}-\mathbf{r}[k]-\mathbf{c}[k])^T(\Gamma+\Sigma_k)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}[k]-\mathbf{c}[k]) \tag{5.17}$$

With these approximations, the estimate in (5.11) has the form

$$\hat{\mathbf{x}}_{MMSE} = \sum_{k=1}^{K-1} f[k]\,\mathbf{b}[k] \tag{5.18}$$

where

$$f[k] = \frac{\dfrac{P[k]}{|\Gamma+\Sigma_k|^{1/2}}\exp(-d[k]/2)}{\displaystyle\sum_{l=0}^{K-1}\dfrac{P[l]}{|\Gamma+\Sigma_l|^{1/2}}\exp(-d[l]/2)} \tag{5.19}$$

where $f[k]$ has the interpretation of the *a posteriori* probability of mixture $k$ given the acoustic information $\mathbf{z}$ for the environmental parameters $\mathbf{n}$ and $\mathbf{q}$, and $\mathbf{b}[k]$ is the conditional mean for mixture component $k$.

Alternatively, (5.18) can be expressed as

$$\hat{\mathbf{x}}_{MMSE} = \mathbf{z} - \mathbf{w} \tag{5.20}$$

where the correction vector $\mathbf{w}$ is given by

$$\mathbf{w} = \sum_{k=0}^{K-1} f[k]\,[\Sigma_k(\Sigma_k+\Gamma)^{-1}(\mathbf{q}+\mathbf{r}[k]) + \Gamma(\Sigma_k+\Gamma)^{-1}(\mathbf{z}-\mathbf{c}[k])] \tag{5.21}$$

so that Equation (5.20) resembles the SDCN algorithm. In this case, however, the correction $\mathbf{w}$ is a weighted sum of *codeword-dependent* corrections.

## 5.3. ML Estimation of Noise and Spectral Tilt

In this section we develop the ML estimates for the noise **n** and equalization function **q** from an ensemble of $N$ cepstrum vectors $Z = \mathbf{z}_0,\ldots,\mathbf{z}_{N-1}$. This maximization is done using only the acoustic information for simplicity in the computations. Since no closed-form solution is available, we will present an iterative algorithm.

If no *a priori* information is given about the noise and equalization vectors **n** and **q**, the optimum estimation method is maximum likelihood. The parameters **n** and **q** are chosen to maximize

$$p(Z|\mathbf{n},\mathbf{q}) \tag{5.22}$$

By assuming that different frames are independent from each other, we can use the expression

$$\ln\ p(Z|\mathbf{n},\mathbf{q}) = \sum_{i=0}^{N-1} \ln\ p(\mathbf{z}_i|\mathbf{n},\mathbf{q}) \tag{5.23}$$

whose maximization leads to

$$\nabla_n \ln\ p(Z|\mathbf{n},\mathbf{q}) = \sum_{i=0}^{N-1} \frac{\nabla_n\, p(\mathbf{z}_i|\mathbf{n},\mathbf{q})}{p(\mathbf{z}_i|\mathbf{n},\mathbf{q})} = \mathbf{0} \tag{5.24}$$

where a similar expression can be derived for the gradient of **q**, where $p(\mathbf{z}_i|\mathbf{n},\mathbf{q})$ can be expressed according to Equation (5.9) as follows:

$$p(\mathbf{z}_i|\mathbf{n},\mathbf{q}) = \alpha\frac{P_i[0]}{|\Gamma|^{1/2}}\ \exp(-\frac{1}{2}(\mathbf{z}_i-\mathbf{n}-\mathbf{s}[0])\,\Gamma^{-1}(\mathbf{z}_i-\mathbf{n}-\mathbf{s}[0]))$$

$$+\ \alpha\sum_{k=1}^{K-1}\frac{P_i[k]}{|C_k|^{1/2}}\ \exp(-\frac{1}{2}(\mathbf{z}_i-\mathbf{q}-\mathbf{r}[k]-\mathbf{c}[k])\,C_k^{-1}(\mathbf{z}_i-\mathbf{q}-\mathbf{r}[k]-\mathbf{c}[k])) \tag{5.25}$$

where $\alpha$ is a constant factor. The first term is expressed as a function of the noise **n** explicitly to reflect the fact that the noise codeword ($k = 0$) is largely insensitive to **q**, and depends mostly on **n**. Similarly, the other codewords are largely insensitive to **n** and depend mostly on **q**.

Since (5.24) and (5.25) lead to a highly nonlinear equation, we use a variant of the EM algorithm described in Appendix F. Instead of maximizing the likelihood directly

$$L(\mathbf{n},\mathbf{q}) = \sum_{i=0}^{N-1} \ln f_{\mathbf{z}}(\mathbf{z}_i;\mathbf{n},\mathbf{q}) \tag{5.26}$$

we can maximize a function $U$ instead

$$U(\mathbf{n},\mathbf{q},\mathbf{n}',\mathbf{q}') = \sum_{i=0}^{N-1} E\{\ln f_{\mathbf{xz}}(\mathbf{x}_i,\mathbf{z}_i;\mathbf{n},\mathbf{q})|\mathbf{z}_i;\mathbf{n}',\mathbf{q}'\} \tag{5.27}$$

that is the expected value of the logarithm of the *joint* density of the observed data $Z$, the unobserved data $X$ and the correction vectors $\{\mathbf{r}(\mathbf{x}_i)\}$ given an estimate of the environmental parameters **n'** and **q'**. With these definitions, the function $U$ in Equation (5.27) has the form (See Appendix G):

$$U = \sum_{i=0}^{N-1} \frac{\sum_{k=0}^{K-1} P_i[k] \int p(\mathbf{z}_i | \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}', \mathbf{q}') p(\mathbf{x} | k) \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}, \mathbf{q}) \, d\mathbf{x}_i}{\sum_{k=0}^{K-1} P_i[k] \int p(\mathbf{z}_i | \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}', \mathbf{q}') p(\mathbf{x}_i | k) \, d\mathbf{x}_i} \tag{5.28}$$

that is given as a function of the old parameter vectors $\mathbf{n}'$ and $\mathbf{q}'$. The maximization of (5.28) gives us new estimates for $\hat{\mathbf{n}}$ and $\hat{\mathbf{q}}$ that can be taken as initial values for the next iteration. The new estimates have the form (See Appendix G)

$$\hat{\mathbf{n}} = \frac{\sum_{i=0}^{N-1} f_i[0] \, \mathbf{z}_i}{\sum_{i=0}^{N-1} f_i[0]} \tag{5.29}$$

$$\hat{\mathbf{q}} = \frac{\sum_{i=0}^{N-1} \sum_{k=1}^{K-1} f_i[k] \, [\mathbf{z}_i - \mathbf{c}[k] - \mathbf{r}'[k]]}{\sum_{i=0}^{N-1} \sum_{k=1}^{K-1} f_i[k]} \tag{5.30}$$

It is important to note that in this case, due to the quadratic form of $U$ with both $\mathbf{n}$ and $\mathbf{q}$, there is just one maximum in the function. That is, the local and global maxima in this case are the same, under the assumptions of the model. In practice, different starting points always converged to the same point for some cases analyzed in detail. This concludes the derivation of the CDCN algorithm. In the following section we will summarize the results.

## 5.4. Summary of the CDCN Algorithm

We present here a summary of the CDCN algorithm with its two steps: estimation of the environmental parameters **n** and **q**, and estimation of the compensated speech given **n** and **q**.

---

**ENVIRONMENT ADAPTATION** achieved by ML estimation of the environmental parameters **n** and **q** is done as follows:

1. Assume initial values of $\hat{\mathbf{n}}^{(0)}$ and $\hat{\mathbf{q}}^{(0)}$ for $j = 1$.
2. **Estimate** the correction vectors $\hat{\mathbf{r}}[k]$ for $k = 0, 1, \ldots, K-1$; given $\hat{\mathbf{n}}^{(j-1)}$, $\hat{\mathbf{q}}^{(j-1)}$, and $\mathbf{x} = \mathbf{c}[k]$ according to (4.10):

$$\hat{\mathbf{r}}^{(j)}[k] = IDFT\,\{\ln\,(1 + e^{DFT\,[\hat{\mathbf{n}}^{(j-1)} - \hat{\mathbf{q}}^{(j-1)} - \mathbf{c}[k]]})\} \tag{5.31}$$

   and the *a posteriori* probabilities for the mixtures $f_i[k]$ as

$$f_i[k] = \frac{\dfrac{P_i[k]}{|C_k|^{1/2}}\exp(-d_i[k]/2)}{\sum_{l=0}^{K-1}\dfrac{P_i[l]}{|C_l|^{1/2}}\exp(-d_i[l]/2)} \qquad \begin{cases} k = 0, 1, \ldots, K-1 \\ i = 0, 1, \ldots, N-1 \end{cases} \tag{5.32}$$

   where the distances $d_i[k]$ are given by

$$d_i[k] = \mathbf{e}_i^T[k]\,C_k^{-1}\,\mathbf{e}_i[k] \tag{5.33}$$

   and the error vectors are

$$\mathbf{e}_i[k] = \mathbf{z}_i - \hat{\mathbf{q}}^{(j-1)} - \hat{\mathbf{r}}^{(j)}[k] - \mathbf{c}[k]) \tag{5.34}$$

3. **Maximize** the log-likelihood (5.23). The new estimates for $\hat{\mathbf{n}}^{(j)}$ and $\hat{\mathbf{q}}^{(j)}$ are

$$\hat{\mathbf{n}}^{(j)} = \frac{\sum_{i=0}^{N-1} f_i[0]\,\mathbf{z}_i}{\sum_{i=0}^{N-1} f_i[0]} \tag{5.35}$$

$$\hat{\mathbf{q}}^{(j)} = \frac{\sum_{i=0}^{N-1}\sum_{k=1}^{K-1} f_i[k]\,[\mathbf{z}_i - \mathbf{c}[k] - \hat{\mathbf{r}}^{j}[k]]}{\sum_{i=0}^{N-1}\sum_{k=1}^{K-1} f_i[k]} \tag{5.36}$$

4. Stop if convergence has been reached obtaining $\hat{\mathbf{n}}$ and $\hat{\mathbf{q}}$; otherwise go to Step 2.

**MMSE ESTIMATION** of $\mathbf{x}_i$ Once the ML estimation of **n** and **q** has converged, for every frame in the utterance, the clean speech vectors are estimated as:

$$\hat{\mathbf{x}}_i = \mathbf{z}_i - \hat{\mathbf{q}} - \sum_{k=1}^{K-1} f_i[k]\,\hat{\mathbf{r}}[k] \quad i = 0, 1, \ldots, N-1; \tag{5.37}$$

---

Let's attempt to find an interpretation for Equations (5.35), (5.36) and (5.37). Equation (5.35) basically estimates the noise vector $\mathbf{n}$ by a weighted sum of all the vectors $\{\mathbf{z}_i\}$ with the weights representing the *a posteriori* probability of each vector being noise. Equation (5.36) estimates $\mathbf{q}$ as a weighted sum of the deviations between the vector $\mathbf{z}_i$ and the corresponding "corrected" codeword $\mathbf{c}[k] + \hat{\mathbf{r}}[k]$. The weights $f_i[k]$ represent again the *a posteriori* probability for the mixture $k$ at frame $i$, or how close the input vector $i$ is to mixture $k$ in a probabilistic sense given the acoustics. Finally equation (5.37) includes a correction that is a sum of all the correction vectors weighted by the appropriate *a posteriori* probabilities.

## 5.5. Implementation Details

In implementing the CDCN algorithm in SPHINX there are a number of parameters that were chosen. For simplicity, all the covariance matrices $C_k$ are assumed to be equal to $\sigma^2 \mathbf{I}$, so that we can continue to use the Euclidean distance. Although this is not a valid assumption, it was adopted for expediency. We also assumed that $\Gamma$ equals $\gamma^2 \mathbf{I}$, which is actually not the case when frequency warping is performed. The codebook elements $\{\mathbf{c}[k]\}$ are estimated with a standard Lloyd algorithm in which the CDCN algorithm was embedded. The initial estimate for the noise is the average of frames whose power is below a threshold (the same computation used in the algorithms of Chapter 3). The equalization vector is initialized to zero, although if some *a priori* information is available it could be used as an initial estimate.

In this implementation the codebook contained 128 vectors, and the value for $\gamma$ was set empirically to 0.3. The value of $\sigma$ was obtained in the iteration that reestimated the codebook vectors; the final value for 128 vectors was 0.54.

Furthermore, all $P_i[k]$ are considered identical, except for $P_i[0]$ which is representative of the rate of noise frames in the database. Therefore, $P_i[0]$ was set to 0.25, while $P_i[k] = (1 - P_i[0]) / (K - 1)$ according to Equations (5.8) and (5.9).

## 5.6. Evaluation Results

Table 5-1 describes the recognition accuracy of the original SPHINX system with no preprocessing, and with the SDCN and CDCN algorithms. Use of the CDCN algorithm brings the performance obtained when training on the CLSTK and testing on the CRPZM to the level observed when the system is trained and tested on the CRPZM. Moreover, use of CDCN improves performance obtained when training and testing on the CRPZM to a level greater than the baseline performance.

To compare the CDCN algorithm with the previous algorithms we include the 3D

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% |
| **SDCN** | N/A | 67.2% | 76.4% | 75.5% |
| **CDCN** | 85.3% | 74.9% | 73.7% | 77.9% |

**Table 5-1:** Comparison of recognition accuracy of SPHINX with no processing, SDCN and CDCN algorithms. The system was trained and tested using all combinations of the CLSTK and CRPZM microphones.

spectrograms of the same utterance in Figures 5-2 and 5-3 for the CLSTK and CRPZM respectively. We note that in this case the algorithm is applied to both recordings.

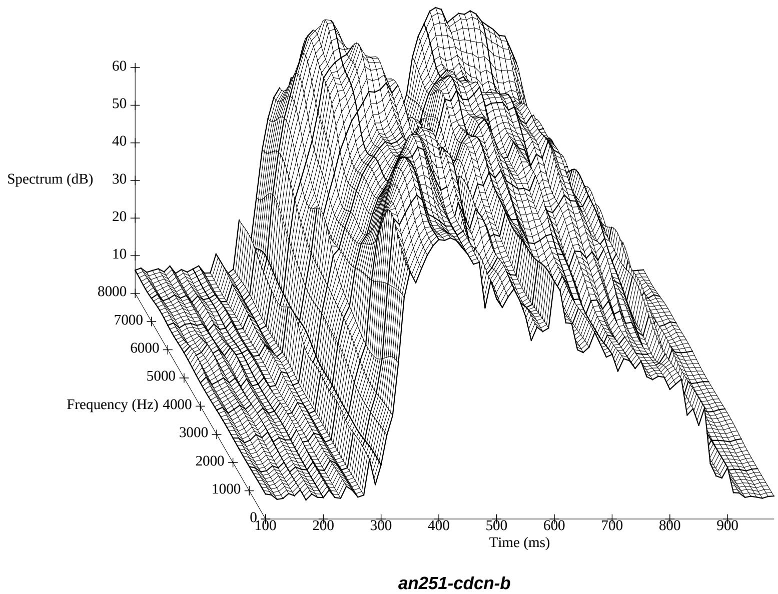



**Figure 5-2:** 3D Spectrogram of the utterance *yes* recorded with the CLSTK microphone processed with the CDCN algorithm.

We can see that the noise level has been reduced greatly not just for the CRPZM but for the CLSTK as well. The tilt in the spectrum has been almost completely eliminated as can be observed by looking at the frame at time 400 ms (the segment with highest SNR in the utterance).

To confirm the ability of the CDCN algorithm to adapt to new environmental conditions, a series of tests was performed with the 5 new stereo speech databases described in Chapter 2. The test data were all collected after development of the CDCN algorithm was completed. In all cases the system was trained using the Sennheiser HMD224. The "second" microphones (with which the system was *not* trained) were:

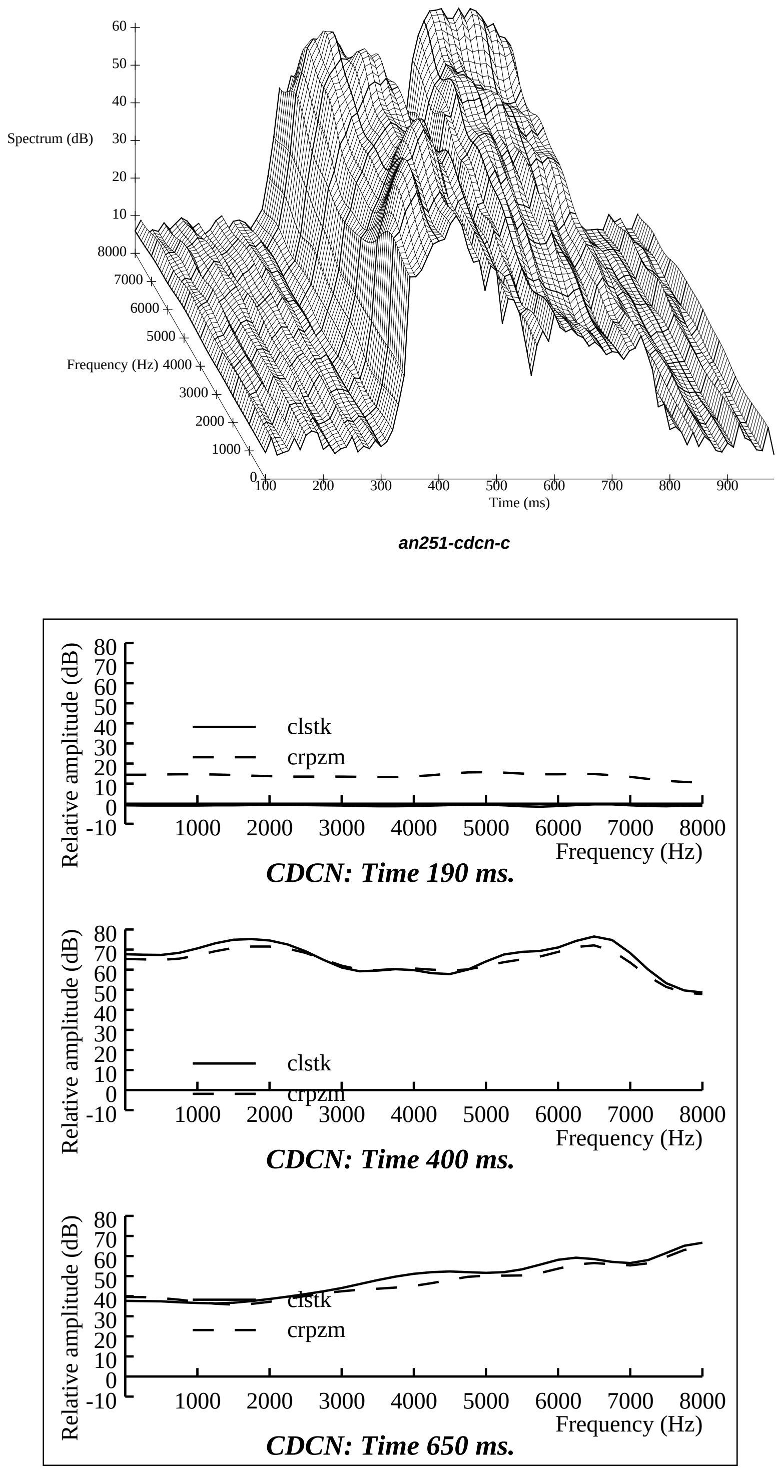



**Figure 5-3:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with CDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

- The Crown PCC160 desk-top phase-coherent cardioid microphone (CRPCC160). (This is the new DARPA "standard" desk-top microphone.)
- An independent test set using the Crown PZM6fs.
- The Sennheiser 518 dynamic cardioid, hand-held microphone (SENN518).
- The Sennheiser ME80 electret supercardioid stand-mounted microphone (SENNME80).
- An HME lavalier microphone that also used an FM receiver (HME).

| TEST | CLSTK | CRPCC160 |
|---|---|---|
| **BASE** | 82.4% | 70.2% |
| **CDCN** | 81.0% | 78.5% |

| TEST | CLSTK | CRPZM6FS |
|---|---|---|
| **BASE** | 84.8% | 41.8% |
| **CDCN** | 83.3% | 73.9% |

| TEST | CLSTK | SENN518 |
|---|---|---|
| **BASE** | 87.2% | 84.5% |
| **CDCN** | 82.2% | 83.3% |

| TEST | CLSTK | SENNME80 |
|---|---|---|
| **BASE** | 83.7% | 71.4% |
| **CDCN** | 81.5% | 80.7% |

| TEST | HME | CRPCC160 |
|---|---|---|
| **BASE** | 55.9% | 56.3% |
| **CDCN** | 81.7% | 72.2% |

**Table 5-2:** Analysis of performance of SPHINX for the baseline and the CDCN algorithm. Two microphones were recorded in stereo in each case. The microphones compared are the Sennheiser HMD224, 518, ME80, the Crown PZM6FS and PCC160, and the HME microphone. Training was done with the Sennheiser HMD224 in all cases.

In Table 5-2 we compare results using the CDCN algorithm to baseline performance. With this algorithm great robustness is obtained across microphones. However, there is a slight drop in performance when training and testing on the Sennheiser HMD224. We believe that one cause for this is that estimates of **q** and **n** are not very good for short utterances.

## 5.7. Summary

Although the SDCN algorithm performed acceptably, it had the drawback that a stereo database was required to train its correction vectors and hence it was not microphone-independent. The CDCN algorithm fits into an environment-independent framework because it estimates the parameters of the current environment via maximum likelihood. This capability of adaptation to the environment makes the CDCN algorithm very attractive. The algorithm has been tested with a number of different microphones and in all cases showed great robustness.

The key information that allowed us to estimate the parameters of the environment, noise and spectral equalization, is the use of the *a priori* information about the speech cepstra contained in a codebook. By observing how that codebook is transformed under some noise **n** and equalization **q**, we were able to see the distribution of vectors under the current environment differed from the transformed distribution.

Another flaw of the SDCN algorithm was that the same correction was applied for all frames with the same SNR while it is clear that different frames would require different corrections. The CDCN algorithm uses corrections that are *codeword-dependent*.

The speech frames processed by the CDCN algorithm exhibit a much lower noise level, and the tilt in the spectrum is insignificant after the transformation. This shows that the model of the environment is quite accurate and that the CDCN algorithm is the first of the algorithms described in this thesis that can be considered to be *environment independent*, because the accuracy of the system when trained on speech recorded with a close-talking microphone and tested with speech recorded with a desk-top microphone is essentially equivalent to our benchmark (the accuracy of the system when trained *and* tested on speech recorded with the desk-top microphone).

# 6 Improving the Efficiency

In this chapter we present two additional algorithms, the Interpolated SDCN and a fixed CDCN, that are substantially more efficient than the CDCN. The ISDCN algorithm, based on the asymptotic properties of the correction vectors, combines the simplicity of the SDCN algorithm while adding environment normalization capabilities. The fixed CDCN algorithm uses a correction that is *codeword-dependent* but like the SDCN algorithm, it has to be recalibrated for every new environment with a stereo database.

## 6.1. Interpolated SDCN

One of the deficiencies of the SDCN algorithm is the inability to adapt to new environments since the correction vectors are derived from a stereo database of our "standard" Sennheiser HMD224 and the new microphone. On the other hand, the algorithm is extremely simple and performs remarkably well. In this section we present the ISDCN algorithm, *Interpolated SNR-Dependent Cepstral Normalization*, as an extension of the SDCN algorithm to environment independence.

The correction vectors of ISDCN will be a function of the instantaneous SNR like the SDCN, and the environmental parameters **n** and **q**. Therefore, the compensated vector $\hat{\mathbf{x}}_i$ has the form

$$\hat{\mathbf{x}}_i = \mathbf{z}_i - \mathbf{w}(\mathbf{n}, \mathbf{q}, SNR_i) \tag{6.1}$$

This explicit dependence of **w** on **n** and **q** will allow us to estimate these parameters from the speech data by using the same criterion used in CDCN: minimization of the difference between the acoustic space of the current utterance and a universal acoustic space characterized by a codebook of cepstral vectors.

Although CDCN accomplishes this by maximum likelihood, we opted to use the minimization of the accumulated VQ distortion for ISDCN. Since SPHINX is already performing Vector Quantization, ISDCN can be implemented with very little computational overhead.

Given an ensemble of $N$ frames, the accumulated VQ distortion has the form

$$D(\mathbf{n},\mathbf{q},k_i) = \sum_{i=0}^{N-1} \| \mathbf{z}_i - \mathbf{w}(\mathbf{n},\mathbf{q},SNR_i) - \mathbf{c}[k_i] \|^2 \tag{6.2}$$

where $k_i$ is the label at frame $i$ that minimizes the distortion

$$\| \mathbf{z}_i - \mathbf{w}(\mathbf{n},\mathbf{q},SNR_i) - \mathbf{c}[k_i] \|^2 \tag{6.3}$$

In ISDCN the correction vector is expressed as

$$\mathbf{w}_i(\mathbf{n},\mathbf{q},SNR) = \mathbf{n} + (\mathbf{q}-\mathbf{n}) f(SNR_i) \tag{6.4}$$

where the function $f$ *interpolates* between the noise $\mathbf{n}$ at low SNR and the equalization vector $\mathbf{q}$ at high SNR, so that the correction vector has the asymptotic characteristics noted in Chapter 4. We selected $f$ to be the sigmoid function

$$f_i(x) = 1/[1+\exp(-\alpha_i x + \beta_i)] \quad \alpha > 0 \tag{6.5}$$

because it satisfies the asymptotic behavior of being $f \approx 0$ at low SNR and $f \approx 1$ at high SNR. It is also monotonic and very smooth.

The noise vector $\mathbf{n}$ can be reliably estimated by averaging a number of noise frames. We basically used the same procedure that we used in the spectral subtraction algorithms in Chapter 3: average all frames whose $\mathbf{c}[0]$ is below a threshold.

Estimation of the equalization vector required the criterion used in Equation (6.2):

1. Start with an initial estimate for $\hat{\mathbf{q}}^{(0)}$ and $j = 1$
2. Label all frames, *i.e.* find the value of $k_i^{(j)}$ that minimizes the distortion

$$\| \mathbf{z}_i - \mathbf{w}(\mathbf{n},\hat{\mathbf{q}}^{(j-1)},SNR_i) - \mathbf{c}[k_i^{(j)}] \|^2 \quad i \le 0 \le N-1 \tag{6.6}$$

3. Estimate $\mathbf{q}^{(j)}$ from all the frames in the utterance:

$$\hat{\mathbf{q}}^{(j)} = \mathbf{n} + \frac{\sum_{i=0}^{N-1} (\mathbf{z}_i - \mathbf{n} - \mathbf{c}[k_i^{(j)}]) f(SNR_i)}{\sum_{i=0}^{N-1} f^2(SNR_i)} \tag{6.7}$$

4. If convergence has been reached stop, else go to step 2.

It can be shown that the labels chosen in Step 2 that minimize Equation (6.6), will also minimize the overall distortion

$$D(\mathbf{n},\hat{\mathbf{q}}^{(j-1)},k_i^{(j)}) \le D(\mathbf{n},\hat{\mathbf{q}}^{(j-1)},k_i^{(j-1)}) \tag{6.8}$$

for a fixed $\mathbf{n}$ and $\hat{\mathbf{q}}^{(j-1)}$. Likewise, it can be shown that the new equalization vector selected by Equation (6.7) in Step 3 will reduce the distortion

$$D(\mathbf{n},\hat{\mathbf{q}}^{(j)},k_i^{(j)}) \le D(\mathbf{n},\hat{\mathbf{q}}^{(j-1)},k_i^{(j)}) \tag{6.9}$$

Therefore, this procedure is guaranteed to converge to a minimum of the accumulated distortion D, as every iteration will decrease the overall VQ distortion.

Ay this point we note that since we are using the codebook of cepstral vectors without the power term (*i.e.* without **c**[0]), the value of **q**[0] cannot be computed by Equations (6.6) and (6.7). The constraint we used to estimate **q**[0], the gain control, was that the dynamic range of the utterance ($\mathbf{c}_{max}[0] - \mathbf{c}_{min}$) had to be constant.

For the evaluation $\alpha_i$ and $\beta_i$ were set empirically to 3.0 for $i > 0$ and 6.0 for $i = 0$, by inspection of the curves in Figure 4-4. The equalization estimate $\hat{\mathbf{q}}$ given by Equation (6.7) exhibited a large variance for short utterances which introduced noise into the system. To ameliorate this problem, we only reestimate the first 4 cepstral coefficients of **q**, setting to 0 the high order ones. This reflects the fact that the equalization vector must be a smooth function. In Table 6-1 we show the performance of the census database with the ISDCN algorithm.

| **TRAIN TEST** | **CLSTK CLSTK** | **CLSTK CRPZM** |
|---|---|---|
| **BASE** | 85.3% | 18.6% |
| **CDCN** | 85.3% | 74.9% |
| **ISDCN** | 84.8% | 62.1% |

**Table 6-1:** Performance of the ISDCN algorithm as compared with the baseline and the CDCN. The algorithm was applied to both CLSTK and CRPZM and training was done with the processed CLSTK.

Since ISDCN will estimate the environmental parameters **n** and **q** for every utterance, it does not have to be recalibrated for every new environment. An evaluation with the microphone recordings used in Chapter 5 is presented in Tables 6-2 and 6-3. We see that the algorithm improves over the baseline in the cross conditions, although not as much as the CDCN for recordings with low SNR. For the case of CLSTK testing, the ISDCN algorithm does on the average about the same as the baseline, and slightly better than the CDCN algorithm.

## 6.2. Fixed CDCN

In this section we describe the Fixed CDCN algorithm that combines some of the attractive features from both the SDCN and CDCN algorithms. The motivation for this algorithm is to obtain an algorithm that as accurate as CDCN and as computationally efficient as SDCN.

| TEST | CLSTK | CRPZM6FS |
|---|---|---|
| **BASE** | 84.8% | 41.8% |
| **CDCN** | 83.3% | 73.9% |
| **ISDCN** | 86.1% | 73.7% |
| **FCDCN** | N/A | 79.3% |

| TEST | CLSTK | CRPCC160 |
|---|---|---|
| **BASE** | 82.4% | 70.2% |
| **CDCN** | 81.0% | 78.5% |
| **ISDCN** | 82.3% | 75.4% |
| **FCDCN** | N/A | 77.1% |

| TEST | CLSTK | SENN518 |
|---|---|---|
| **BASE** | 87.2% | 84.5% |
| **CDCN** | 82.2% | 83.3% |
| **ISDCN** | 87.2% | 83.5% |
| **FCDCN** | N/A | 83.4% |

| TEST | CLSTK | SENNME80 |
|---|---|---|
| **BASE** | 83.7% | 71.4% |
| **CDCN** | 81.5% | 80.7% |
| **ISDCN** | 83.2% | 78.5% |
| **FCDCN** | N/A | 81.1% |

**Table 6-2:** Analysis of performance of SPHINX for the baseline and the CDCN, ISDCN and FCDCN algorithms. Two microphones were recorded in stereo in each case. The microphones compared are the Sennheiser HMD224 (CLSTK), Crown PZM6FS, Crown PCC160, Sennheiser 518 and SennheiserME80. Training was done with the Sennheiser HMD224 (CLSTK) from the census database in all cases. The correction vectors for the FCDCN were estimated from each stereo database and are different for every experiment.

The Fixed CDCN applies a correction that depends on the instantaneous SNR of the input, like SDCN, and this correction is different for every codeword, like the CDCN:

$$\hat{\mathbf{x}} = \mathbf{z} + \mathbf{r}[k, SNR] \tag{6.10}$$

| TEST | HME | CRPCC160 |
|---|---|---|
| **BASE** | 55.9% | 56.3% |
| **CDCN** | 81.7% | 72.2% |
| **ISDCN** | 76.6% | 69.7% |

**Table 6-3:** Analysis of performance of SPHINX for the baseline and the CDCN and ISDCN algorithms. The Crown PCC160 and the HME FM microphone were recorded in stereo using training with the Sennheiser HMD224 (CLSTK) from the census database.

The selection of the appropriate codeword is done at the VQ stage, so that label $k$ is chosen to minimize

$$\|\mathbf{z} + \mathbf{r}[k, SNR] - \mathbf{c}[k]\|^2 \tag{6.11}$$

This technique will be applied only to the CRPZM speech so as to make it as close as possible to the CLSTK speech. We will describe a method for training these correction vectors using the EM algorithm. The incomplete data consist of the noisy speech, and the complete data include the clean speech, as it is known in the stereo database.

### 6.2.1. Estimating the Correction Vectors

In this subsection we develop the estimation formulae for the correction vectors $\mathbf{r}[k, SNR]$ and the variances $\sigma^2[SNR]$ via the EM algorithm. Briefly, the algorithm labels the CRPZM speech without any knowledge of the CLSTK speech according to Equations (6.10) and (6.11). After this labeling is completed the correction vectors are reestimated as the ones that minimize the differences between the CLSTK and the CRPZM cepstra on a frame by frame basis.

The densities to be used in the EM algorithm are defined here. The *pdf* of the CLSTK speech given mixture $k$ is

$$p(\mathbf{x}|k) = \frac{C}{\sigma} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{c}[k]\|^2\right) \tag{6.12}$$

The CRPZM speech is modeled as a Gaussian random vector

$$p(\mathbf{z}|\mathbf{x}, k, \mathbf{r}', SNR = l) = \frac{C'}{\sigma[l]} \exp\left(-\frac{1}{2\sigma^2[l]} \|\mathbf{z} + \mathbf{r}'[k, l] - \mathbf{x}\|^2\right) \tag{6.13}$$

if the value of $\mathbf{x}$ is known, and as

$$p(\mathbf{z}|k, \mathbf{r}', SNR = l) = \frac{C'}{\sigma[l]} \exp\left(-\frac{1}{2\sigma^2[l]} \|\mathbf{z} + \mathbf{r}'[k, l] - \mathbf{c}[k]\|^2\right) \tag{6.14}$$

if no knowledge is available on $\mathbf{x}$. In (6.13) and (6.14), $\mathbf{r}'[k, l]$ is the correction vector when the true mixture is $k$ and the input SNR of $\mathbf{z}$ is $l\Delta_{SNR}$. The variance $\sigma[l]$ is a function of the instantaneous SNR too. With these assumptions, the *a posteriori probability* $p(k|\mathbf{z}_i, \mathbf{r}')$, that we will denote by $f_i[k]$ can be expressed as

$$f_i[k] = \frac{\exp\left(-\frac{1}{2\sigma^2[l_k]}\|\mathbf{z}_i + \mathbf{r}'[k, l_k] - \mathbf{c}[k]\|^2\right)}{\sum_{p=0}^{K-1} \exp\left(-\frac{1}{2\sigma^2[l_p]}\|\mathbf{z}_i + \mathbf{r}'[p, l_p] - \mathbf{c}[p]\|^2\right)} \tag{6.15}$$

where we have assumed that the *a priori* probabilities for the mixtures are identical. We can also obtain the logarithm of the complete data as

$$\begin{aligned} \ln p(\mathbf{z}_i, \mathbf{x}_i, k|\mathbf{r}) &= C'' - \ln\sigma[l_k] - \frac{1}{2\sigma^2[l_k]}\|\mathbf{z}_i + \mathbf{r}[k, l_k] - \mathbf{x}_i\|^2 \\ &- \ln\sigma - \frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{c}[k]\|^2 \end{aligned} \tag{6.16}$$

Instead of maximizing the likelihood $p(\mathbf{z}_i, \mathbf{x}_i, |\mathbf{r})$ directly, we will use the EM algorithm described in Appendix F and maximize the function U instead. In this case the function U has the form

$$U(\mathbf{r}, \mathbf{r}') = \sum_{i=0}^{N-1}\sum_{k=0}^{K-1} \ln p(\mathbf{z}_i, \mathbf{x}_i, k|\mathbf{r})\, p(k|\mathbf{z}_i, \mathbf{r}') \tag{6.17}$$

The correction vectors $\mathbf{r}[k, l_k]$ that maximize Equation (6.17) can be obtained by using Equations (6.15) and (6.16):

$$\mathbf{r}[k, l] = \frac{\sum_{i=0}^{N-1} (\mathbf{x}_i - \mathbf{z}_i)\, f_i[k]\, \delta[SNR_i - l\Delta_{SNR}]}{\sum_{i=0}^{N-1} f_i[k]\, \delta[SNR_i - l\Delta_{SNR}]} \tag{6.18}$$

and the corresponding $\sigma[l]$ given by

$$\sigma^2[l] = \frac{\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} \|\mathbf{x}_i - \mathbf{z}_i - \mathbf{r}[k, l]\|^2 f_i[k]\, \delta[SNR_i - l\Delta_{SNR}]}{\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} f_i[k]\, \delta[SNR_i - l\Delta_{SNR}]} \tag{6.19}$$

The new estimates $\mathbf{r}[k, l]$ and $\sigma[l]$ obtained by Equations (6.18) and (6.19) are guaranteed to increase the likelihood $p(Z, X|\mathbf{r})$. In practice, the algorithm reaches convergence after 2 or 3 iterations. Figure 6-1 shows the resulting variances $\sigma^2[l]$ obtained after the process for $\Delta_{SNR} = 1\,dB$. The large variance exhibited at low SNR reflects the higher uncertainty in the value of the CLSTK speech given the CRPZM speech that occurs at low SNRs.

We have shown in Chapter 5 that what determines the values of the correction vectors is not just $\mathbf{q}$ but $\mathbf{q} - \mathbf{n}$, the difference between the equalization and noise vectors. Therefore in this implementation the first step is to remove the noise vector from all the input frames and then apply the correction vectors. In training the correction vectors are obtained the same way, subtracting the noise vector first. The noise vector is estimated via an EM procedure as described by Van Compernolle [10], in which two Gaussian densities are fitted to the data, one for the noise event and another one for the speech event. In Figure 6-2 we show the sample utterance for the CRPZM when processed by this algorithm. Table 6-4 shows the results of this algorithm.

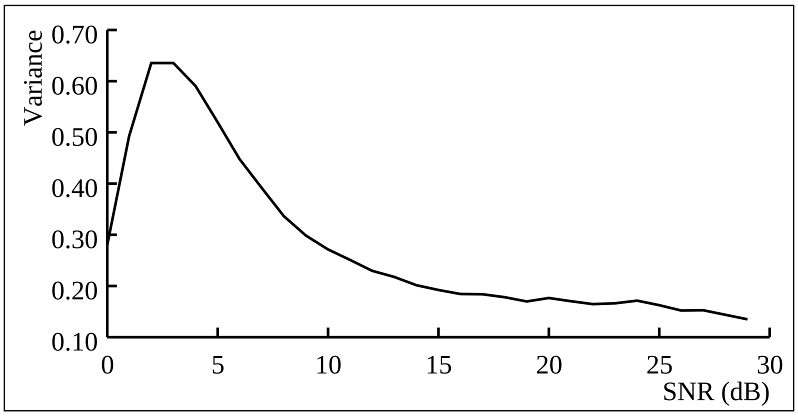



**Figure 6-1:** Variance of the difference vector between the CLSTK and the restored CRPZM speech for different input SNR of the CRPZM.

Another possibility for estimating a similar set of correction vectors was suggested by Gish *et al.* [28]. He used a procedure similar to Equation (6.18) for estimating the corrections but they derived the labels from the CLSTK speech instead of the CRPZM speech. We tried this procedure and obtained a restored speech that exhibited a large amount of *musical noise* as was observed using the PSUB algorithm (See Chapter 3). Many frames in the clean acoustic space are mapped into essentially the same frame in the noisy space because they are masked by the noise, so inverting the process makes many frames that are similar in the noisy acoustic space to become wildly different in the transformed space if this approach is taken. In our criterion the distortion measure is taken in the noisy space, so that small differences in the noisy speech gets translated into small differences in the restored speech.

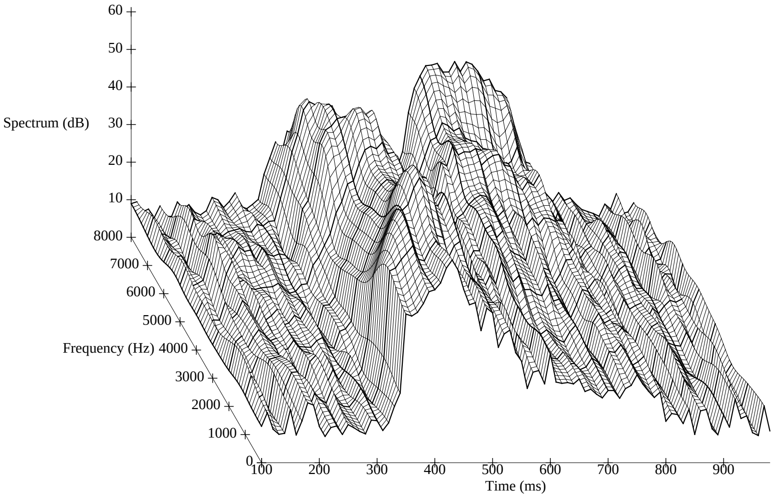
Another variant that we tried for estimating the correction vectors is using the assumption that $\mathbf{x}$ has the distribution

$$p(\mathbf{x}) = \max\ P[k]\ N_{\mathbf{x}}(\mathbf{c}[k], \Sigma_k) \tag{6.20}$$

where we have substituted the sum with the maximum operator. The *a posteriori* probability $p(k|\mathbf{z}_i, \mathbf{r}')$ will be a delta function $\delta[k]$ where $k$ is the codeword that minimizes

$$\|\mathbf{z}_i - \mathbf{c}[k] - \mathbf{r}[k, l]\|^2 \tag{6.21}$$

Using the complete probability of Equation (6.16), our EM algorithm yields the same reestimation formula for $\mathbf{r}[k, l]$ given in Equation (6.18), with $f_i[k]$ being 1 if $k$ is the label for frame $i$ and 0 otherwise. The recognition rate for the CRPZM training on the CLSTK is 72.6%, slightly lower than with the previous training algorithm.




*an251-fcdcn-c*

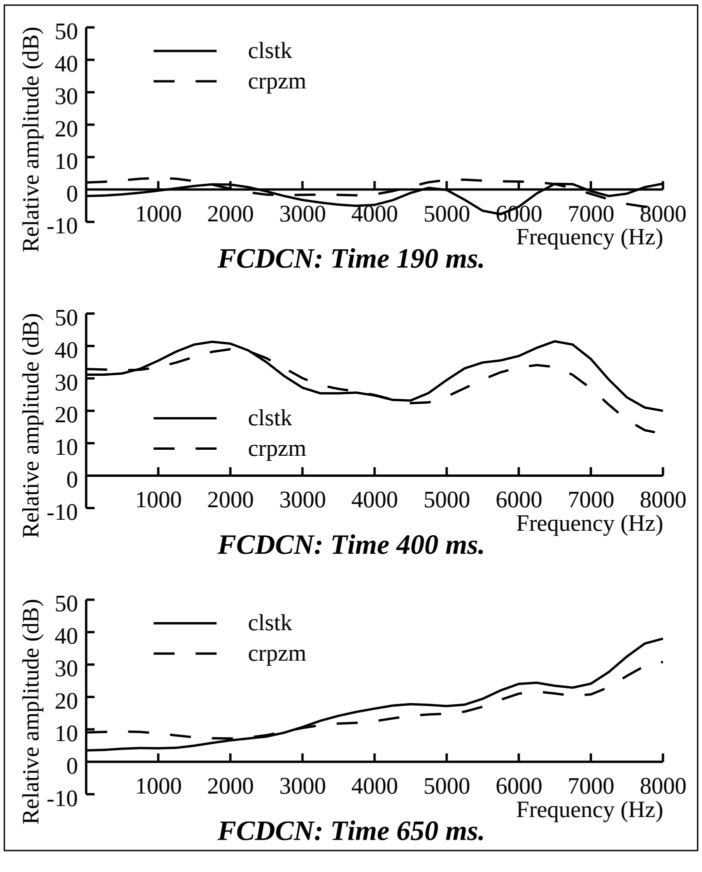



**Figure 6-2:** 3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with FCDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM |
|---|---|---|
| **BASE** | 85.3% | 18.6% |
| **CDCN** | 85.3% | 74.9% |
| **FCDCN** | N/A | 73.1% |

**Table 6-4:** Performance of the Fixed CDCN algorithm as compared with the baseline and the CDCN. This algorithm is only applied to the CRPZM using the training models for the baseline case.

The computational complexity of this fixed CDCN is very low, because the correction vectors are precomputed. However, it does not have the adaptation capabilities of the ISDCN or CDCN. It would be desirable to obtain an algorithm that is computationally efficient and adapts to the environment like the ISDCN but is more immune to noise. Having different sets of correction vectors for different environments (*i.e.* quantizing the environments) would be a possibility, but it would require a large number of different microphone recordings and will be explored in future work.

## 6.3. Estimating the Environmental Parameters from Previous Utterances

In this section we analyze some issues related to real-time implementation. Up to this point, the estimates of the noise **n** and equalization vector **q** for one given utterance were estimated from that same utterance. In a real application this will not be practical. Assuming that these parameters do not change very rapidly with time (quasi-stationarity), one should be able to use the estimates from the previous utterance for the next utterance.

Another test set was collected in stereo with the Sennheiser HMD224 and the CRPZM 6fs. In Table 6-5 we show the baseline performance for this set as well as the FCDCN and ISDCN algorithms. We see that this test set is probably the hardest we have encountered since the performance for the case of training and testing on the CRPZM is only 66.6%. The accuracy of the FCDCN algorithm with the correction vectors computed from the census database exceeds that value.

Inspection of Table 6-5 tells us that using the estimates of the environment computed from the previous utterance (ISDCNprev) yields essentially the same recognition accuracy than if the environmental parameters were estimated from the current utterance (ISDCN) as we have done until now. This fact can be used in implementing a real-time system where the estimates of **n** and **q** are continuously updated. More research is needed on the time constant or number of speech frames used in the estimation of the environmental parameters.

| TRAIN<br>TEST | CLSTK<br>CLSTK | CLSTK<br>CRPZM | CRPZM<br>CLSTK | CRPZM<br>CRPZM |
|---|---|---|---|---|
| **BASE** | 82.4% | 21.3% | 36.6% | 66.6% |
| **FCDCN** | 82.4% | 67.2% | - | - |
| **ISDCN** | 80.8% | 57.1% | - | - |
| **ISDCNprev** | 80.7% | 55.8% | - | - |

**Table 6-5:** Performance of the ISDCN algorithm when the environmental parameters are computed from the same utterance (ISDCN) or the previous utterance (ISDCNprev). The performance is compared to the baseline (BASE) and the FCDCN algorithm.

## 6.4. Summary

In this chapter we have described two algorithms that are very efficient computationally, the ISDCN and the FCDCN algorithm. In both cases they can be implemented with the addition of some book-keeping information at the VQ stage.

The Interpolated SNR-Dependent Cepstral Normalization (ISDCN) algorithm uses correction vectors that interpolate between the noise vector at low SNR to the equalization vector at high SNR. The equalization vector is estimated as the one that minimizes the accumulated VQ distortion. This is possible by the use of an interpolating function that in this case was a sigmoid function.

The Fixed Codeword-Dependent Cepstral Normalization (FCDCN) algorithm uses correction vectors that depend on the instantaneous SNR of the input and are different for every codeword. This algorithm provides a higher accuracy than the CDCN because it is free from some of the assumptions of the CDCN algorithm.

Although more research is needed to investigate how much time is needed for a reliable estimation of the environmental parameters, we have shown that adapting on the previous utterance is feasible for a real-time implementation.

# 7 Frequency Normalization

In this chapter we will discuss the mel-scale cepstral parameters used in SPHINX and some optimizations that we performed on them. Also, an algorithm for frequency normalization based on variable warping of the frequency axis is presented within the framework of minimal VQ distortion. These techniques together decrease the error rate by 15 to 20%.

## 7.1. The Use of Mel-scale Parameters

In this section we introduce the concept of mel-scale cepstral coefficients as the DFT of the logarithm of the power spectral density function over a warped frequency scale. A rationale for its use is presented and the particular implementation of these coefficients in the SPHINX system as a matrix multiplication on LPC cepstrum coefficient is described.

The use of a warped frequency scale as opposed to a linear one has proven advantageous in speech recognition systems. Most researchers agree that frequencies above 4 kHz contribute much less to speech intelligibility and recognition accuracy than frequencies below 4 kHz. Since for systems like SPHINX the sampling rate is 16 kHz, using a linear scale would give the same weight to low and high frequencies, which is undesirable.

Further evidence for the use of a non-linear frequency axis can be extracted from the behavior of the human auditory system. Zwicker [84] introduced the *Bark scale*[27] as an approximation to the discrimination power of different frequencies in the human auditory system.

---

[27]The *Bark scale* is approximately linear for frequencies below 1000 Hz and logarithmic above that level.

Davis and Mermelstein [13] found that *mel-scale* coefficients resulted in improved accuracy. They used a filterbank in which the spacing among the filters and their bandwidths approximated the Bark scale. Researchers at BBN (Chow *et al.* [8]), SRI (Murveit and Weintraub [55]), MIT Lincoln Labs (Paul [65]) and some other laboratories have successfully used cepstral coefficients derived from a mel-scale filterbank approach as the front-end for their speech recognizers.

The SPHINX system also uses mel-scale cepstral coefficients as the parameter set, but they are based on an LPC analysis as opposed to filterbank. Shikano [71] applied the technique of bilinear transformation introduced by Oppenheim and Johnson [61] to warp the frequency axis. More details on the bilinear transform can be found in Appendix C.

Briefly, the bilinear transform is a mapping in the complex plane that maps the unit circle onto itself. It is defined as

$$z_{new}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \quad -1 < \alpha < 1 \tag{7.1}$$

The frequency transformation is obtained by making the substitution $z = e^{j\omega}$ and $z_{new} = e^{j\omega_{new}}$ in (7.1):

$$\omega_{new} = \omega + 2\,\mathrm{arctg}\left[\frac{\alpha \sin(\omega)}{1 - \alpha \cos(\omega)}\right] \tag{7.2}$$

Equation (7.2) is plotted in Figure 7-1 for different values of the warping parameter $\alpha$.

For values of $\alpha$ between 0.4 and 0.8, the warping transformation is similar to the Bark scale. In the SPHINX system the parameter $\alpha$ was set to 0.6. Although SPHINX was not very sensitive to the particular value, 0.6 turned out to be optimum in our evaluations.

As can be seen in Appendix C, the relationship between the cepstral coefficients before warping $\mathbf{c}$ and after the bilinear transform $\mathbf{c}_w$ can be expressed as a matrix multiplication operation

$$\mathbf{c}_w = \mathbf{L}(\alpha)\,\mathbf{c} \tag{7.3}$$

where the warping matrix $\mathbf{L}$ is a function of $\alpha$. In the original signal processing in SPHINX both $\mathbf{c}$ and $\mathbf{c}_w$ had 12 coefficients.

## 7.2. Improving the Frequency Resolution

In this section we analyze the the frequency warping used in the standard SPHINX system and its effect on frequency resolution. Increasing the number of cepstral coefficients before the bilinear transform yielded a moderate decrease in error rate.

The truncation used in the cepstrum vector has to be done so that important information is not lost. It is shown in Appendix B that the LPC-cepstrum is an infinite

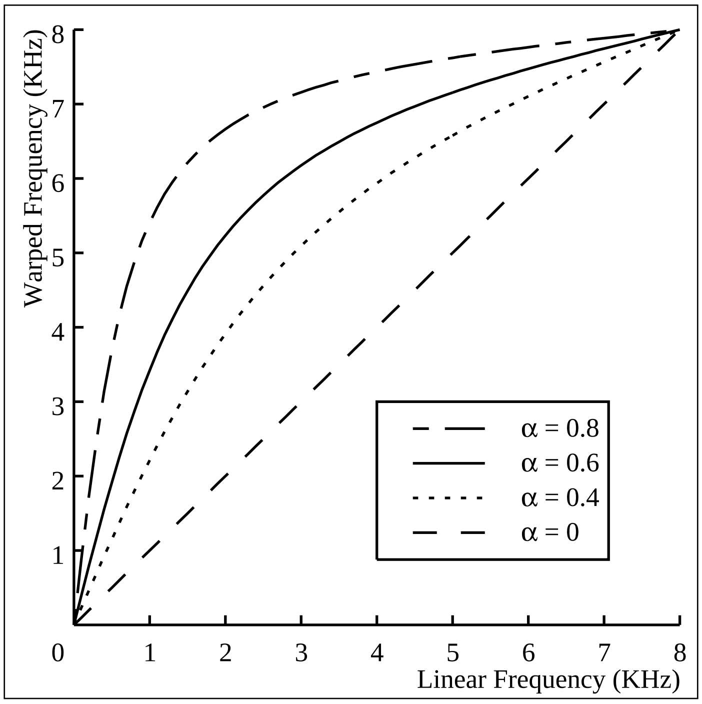



**Figure 7-1:** Frequency mapping of the Bilinear Transform for different values of $\alpha$. Note that $\alpha = 0$ is equivalent to no warping while $\alpha = 0.8$ is a very severe warping of the frequency axis.

sequence. In Appendix C we show that the number of coefficients after the bilinear transform will be infinite. This is to say that both **c** and $\mathbf{c}_w$ in Equation (7.3) will be of infinite dimension. However, since the higher order cepstrum coefficients tend to zero rapidly, the dimension of vector $\mathbf{c}_w$ can be truncated with negligible error. In SPHINX the dimension of $\mathbf{c}_w$ was set to 12 (plus the zeroth order term).

We found that the dimension of **c** had to be larger than 12 in order not to lose frequency resolution. In Figure 7-2 we show the spectral representation of the vowel /ih/ in the word "six". We observe that the use of 12 coefficients for **c** results in a loss of resolution, that is maintained if 32 coefficients are used. With 12 cepstral coefficients, resonances whose bandwidths are smaller that $8000 / 12 = 666$ Hz will not be represented accurately because of the low-pass *liftering*. We observed that the use of 32 coefficients provided a higher low-pass *quefrency* that allowed resonances with bandwidths larger than $8000 / 32 = 250$ Hz to be retained. A dimension of 12 for $\mathbf{c}_w$ was considered sufficient because of stretching of the low frequencies, where the important resonances occur.

We trained and tested SPHINX on the census database using both 12 and 32 coefficients for **c**. The results in Table 7-1 show a 7% decrease in error rate by using 32 coefficients for **c**. This reduction in error rate is consistent with what was observed in the

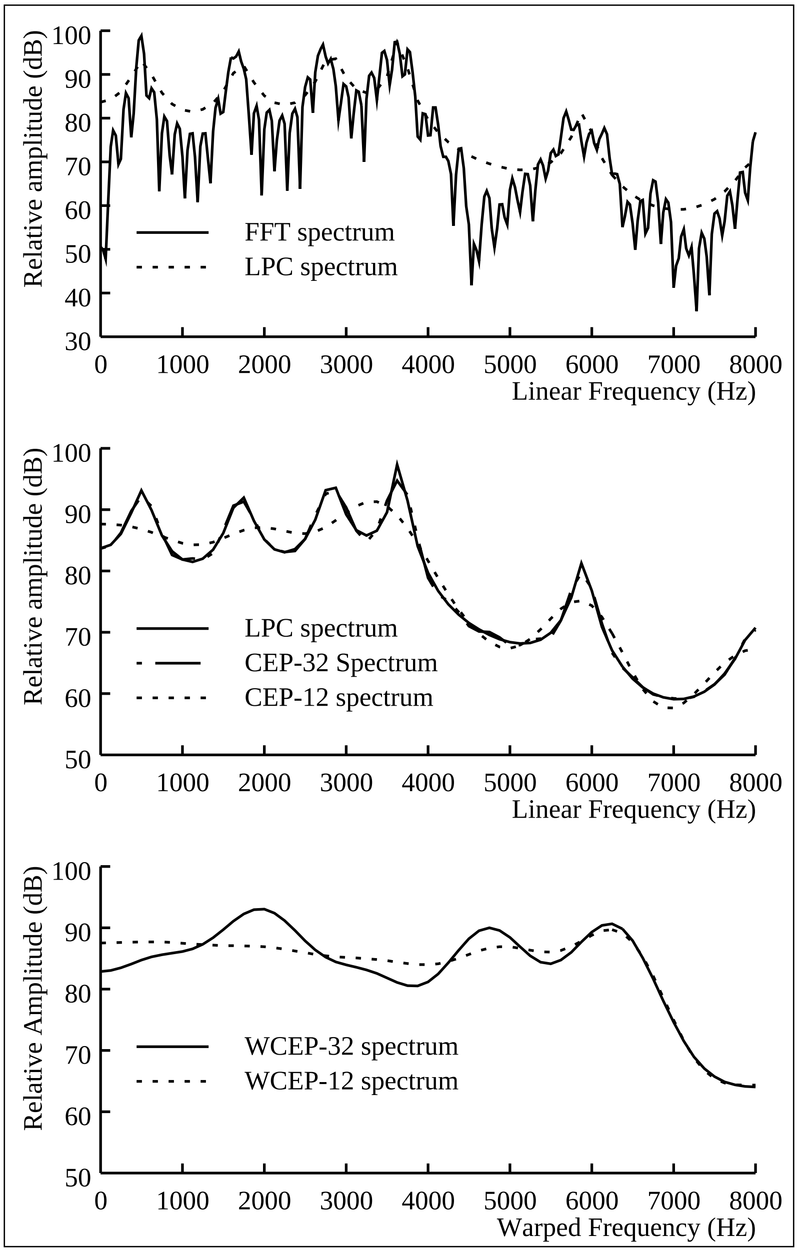



**Figure 7-2:** Spectral analysis of the vowel /ih/. The first graph shows the FFT spectrum and the LPC spectrum of order 14. The second graph shows the LPC spectrum and its cepstral approximation with 32 and 12 coefficients. The third graph shows the warped spectrum plotted from 12 cepstral coefficients after the bilinear transform with $\alpha = 0.6$ when both 32 and 12 coefficients are used before the transform. The use of 12 coefficients before the warping removes the formant structure.

resource management task. We considered the version with 32 coefficients as the baseline condition for all the experiments in this thesis.

| | 12 coeff | 32 coeff |
|---|---|---|
| Recognition accuracy | 84.2 % | 85.3 % |

**Table 7-1:** Effect of the use of 12 and 32 cepstral coefficients before the bilinear transform in SPHINX. The number of coefficients after frequency warping was 12 in both cases.

At first, we were somewhat disappointed that there was only a reduction of 7% after having observed the phenomenon in Figure 7-2. The explanation that we have is that although we are not retaining additional formant information by using 32 coefficients for **c**, this information is not so critical anyway if the system is speaker independent. The effect of a crisper frequency structure obtained by the use of 32 coefficients for **c** is diminished by the fact that different speakers have different formants, which would make the HMM distributions broader.

## 7.3. Variable Frequency Warping

In this section we present a novel technique for frequency normalization based on the use of the bilinear transform with a variable warping parameter. This technique is aimed at closing the gap in performance that exists between speaker-dependent and speaker-independent systems by normalizing the long-term frequency characteristics of different speakers.

Speaker-independent systems perform with an error rate that is about 3 or 4 times greater than similarly trained speaker-dependent systems (Pallett *et al.* [63]). Part of the problem can be found in that speaker-independent systems like SPHINX have to cope with the burden of different formant frequencies of different speakers, that broaden the HMM distributions. It would be desirable to be able to normalize the formant frequencies of different speakers.

It is well known that female speech exhibits higher formant frequencies and pitch than male speech. Furthermore, a source-production model (Rabiner and Schafer [68]) suggests that the nominal resonance frequencies are essentially proportional to the length of the vocal tract. A possible mechanism for normalization would be to warp frequencies more severely for males than for females, so that after this *speaker-dependent* frequency warping, the resonance frequencies coincide.

We propose to achieve the variable frequency warping by using a different $\alpha$ parameter in the bilinear transform for every speaker. The value of $\alpha$ is selected as the one that minimizes the overall VQ distortion, the same criterion used in the last chapter. This algorithm works in an *unsupervised* mode, since it does not require sex information or any other characterization of the speaker's formant frequencies.

For practical considerations we chose to implement the bilinear transform in two stages: the first stage being fixed and the second being variable. This is made possible due to the fact that a cascade of two bilinear transform stages of parameters $\alpha_1$ and $\alpha_2$ is equivalent to another bilinear transform with parameter $(\alpha_1 + \alpha_2)/(1 + \alpha_1\alpha_2)$ as shown in appendix C. The advantage in doing this is that we use the 32 coefficients necessary for the increased frequency resolution only in the first warping stage, while we use only 12 coefficients for the second stage without any loss in accuracy.

Although we could develop a framework for maximization over continuous values of $\alpha$, we choose to work with a discrete grid of values of $\alpha$ with $\alpha = \alpha_0 + i\Delta\alpha$ for simplicity, with $i = -N, \ldots, 0, \ldots, N$.

The new codebook is generated by the Lloyd algorithm used for finding the VQ codebook, with the difference that the $\alpha$ parameter for every speaker is different. This optimum $\alpha$ is the one that minimizes the VQ distortion for all the utterances uttered by a given speaker. Therefore only the labeling is different from the standard Lloyd algorithm.

By doing this we observed that the algorithm always found the highest value for alpha. In the degenerate case of $\alpha = 1$, every input frame is transformed into a constant (the DC value) by the bilinear transform, therefore yielding zero VQ distortion. However this trivial transformation would destroy all the information in the signal. This is the first time that we encounter that minimum distortion is not highly correlated with maximum likelihood and minimum probability of error, but quite the opposite. It was observed empirically that higher values of $\alpha_0$ yielded lower distortion, although lower recognition rate.

Since we wanted to keep $\alpha_0 = 0.6$ as our center value, we had to force that constraint directly. We used 20 female speakers and 20 male speakers in training, with each speaker having his or her own $\alpha_i$ and we imposed that the sum of all displacements had to be zero $\Sigma\, \alpha_i = 0$ when labeling the training data.

This variable warping technique was evaluated with an optimum $\alpha$ per speaker. The center value $\alpha_0$ was set to 0.6, with $N = 5$ and $\Delta = 0.02$. The results of the evaluation on the census database are shown in Table 7-2. As we can see, there is a 12% decrease in error rate. Use of a denser grid did not provide any additional benefit. The value of $\alpha_0$ was not very critical, although 0.6 provided the maximum gain.

| | BASELINE | FREQNORM |
|---|---|---|
| Recognition accuracy | 85.3 % | 87.1 % |

**Table 7-2:** Comparison of the performance of the variable frequency warping with the baseline.

In Figure 7-3 we show the distribution of values of $\alpha_{opt}$ for male and female speakers. As we had anticipated there is a clear separation between them, which confirms the assumptions of the model.

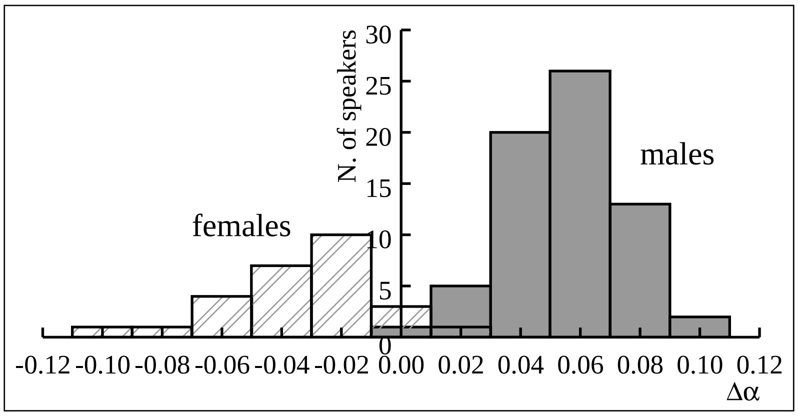



**Figure 7-3:** Histogram of values of $\alpha$ for male and female speakers.

To cross-validate the results in Table 7-2, we used the CLSTK speech from the multi-microphone database. The results of the evaluation are shown in Table 7-3. As can be observed, the algorithm never does worse than the baseline, and on the average there is a 10% decrease in error rate. There is especially a large gain in Set 1 for which the baseline performance was lower than the rest.

| | BASELINE | FREQNORM |
|---|---|---|
| Set 1 | 82.4 % | 85.4 % |
| Set 2 | 84.8 % | 85.4 % |
| Set 3 | 87.2 % | 87.8 % |

**Table 7-3:** Comparison of the performance of the variable frequency warping with the baseline.

## 7.4. Summary

Mel-scale cepstrum coefficients are advantageous for speech recognition especially if a bandwidth larger than 4 kHz is used. SPHINX uses the bilinear transform as a mean of obtaining frequency-warped LPC-cepstrum.

Truncation of the cepstral coefficients is acceptable as long as they retain the frequency structure. After the bilinear transform only 12 coefficients are needed but before the transform 12 is not sufficient. We showed that the use of 32 coefficients before the bilinear transform results in no apparent loss in the accuracy of the spectral representation, and a 7% decrease in error rate.

We proposed the use of the bilinear transform with variable parameter as a means of *frequency normalization*. Histograms of the optimum $\alpha$ show that male and female speech are well separated, with female speech requiring a smaller $\alpha$ (less warping) than male speech. The use of frequency normalization provides an additional decrease in error rate of approximately 10%.

# 8 Summary of Results

In this chapter we summarize the results of this dissertation. Table 8-1 shows the performance of all the algorithms described in this thesis for the census database.

| TRAIN TEST | CLSTK CLSTK | CLSTK CRPZM | CRPZM CLSTK | CRPZM CRPZM | stereo data | Comp. Complex. |
|---|---|---|---|---|---|---|
| **BASE** | 85.3% | 18.6% | 36.9% | 76.5% | no | low |
| **EQUAL** | N/A | 38.3% | 50.9% | 76.5% | no | low |
| **PSUB** | N/A | 38.6% | 70.6% | 70.1% | no | low |
| **MSUB** | N/A | 63.6% | 71.7% | 71.3% | no | low |
| **MMSE1** | N/A | 48.7% | 68.7% | 71.4% | yes | low |
| **EQ+MMSE1** | N/A | 61.4% | 75.8% | 74.3% | yes | low |
| **EQ+MSUB** | N/A | 62.1% | 73.7% | 71.4% | yes | low |
| **MMSEN** | N/A | 66.4% | 75.5% | 72.3% | yes | low |
| **SDCN** | N/A | 67.2% | 76.4% | 75.5% | yes | low |
| **FCDCN** | N/A | 73.1% | 79.3% | 75.8% | yes | low |
| **ISDCN** | 84.8% | 62.1% | 71.4% | 72.4% | no | low |
| **CDCN** | 85.3% | 74.9% | 73.7% | 77.9% | no | medium |

**Table 8-1:** Performance of different normalization algorithms.

The algorithms are the following:

- BASE: Baseline case, no processing (See Chapter 2).
- EQUAL: The CRPZM speech is equalized by adding a fixed cepstral vector that is the difference between the average speech for both CLSTK and CRPZM (See Chapter 3).
- PSUB: The power spectral subtraction rule is applied to the CRPZM speech (See Chapter 3).

- MMSE1: The frequency bands in the CRPZM speech are transformed according to a curve that minimizes the squared error between CLSTK and CRPZM (See Chapter 3).
- MSUB: Magnitude spectral subtraction with under and over-subtraction.  is applied to the CRPZM speech (See Chapter 3).
- EQ+MMSE1: Cascade of EQUAL and MMSE1 for the CRPZM speech (See Chapter 3).
- EQ+MSUB: Cascade of EQUAL and MSUB for the CRPZM speech (See Chapter 3).
- MMSEN: Each frequency band in the CRPZM speech is transformed with a different curve so that the squared error between the CLSTK and CRPZM is minimized (See Chapter 4).
- SDCN: The SDCN algorithm is applied to the CRPZM speech by transforming every cepstral component differently depending on the instantaneous SNR. The goal is to minimize the squared error between CLSTK and CRPZM (See Chapter 4).
- FCDCN: The FCDCN algorithm is applied to the CRPZM speech with a different correction vector for every instantaneous SNR and codeword. Again the goal is to minimize the squared error between CLSTK and CRPZM speech (See Chapter 6).
- ISDCN: The ISDCN algorithm is applied to both CLSTK and CRPZM by estimating directly the noise and equalization vector (See Chapter 6).
- CDCN: The CDCN algorithm is applied to both CLSTK and CRPZM by estimating directly the noise and equalization vector (See Chapter 5).

Since we are mostly concerned with the performance when the system is trained with the CLSTK speech, we show in Figures 8-1, 8-2 and 8-3 the word accuracy of the system trained on the CLSTK and tested on the CRPZM speech for the case of algorithms that attempt independent compensation for noise and spectral tilt, algorithms that do independent versus joint compensation and algorithms that operate in the spectral versus the cepstral domain.

Figure 8-1 shows different compensation schemes in the frequency domain that combat the linear filtering (EQUAL) or additive noise (PSUB, MMSE1, MSUB). MSUB is the one with highest accuracy of all of them.  However, it is still far from the accuracy obtained when the system is trained and tested on the CRPZM microphone.

Figure 8-2 shows a comparison of the performance of algorithms that attempt an independent compensation (cascade of equalization EQUAL and noise suppression MMSE1 and MSUB) and a joint compensation for noise and filtering (MMSEN) in the frequency domain.  We note that the MMSEN algorithm, by using a different transformation curve per frequency, deals better with the colored noise present in the

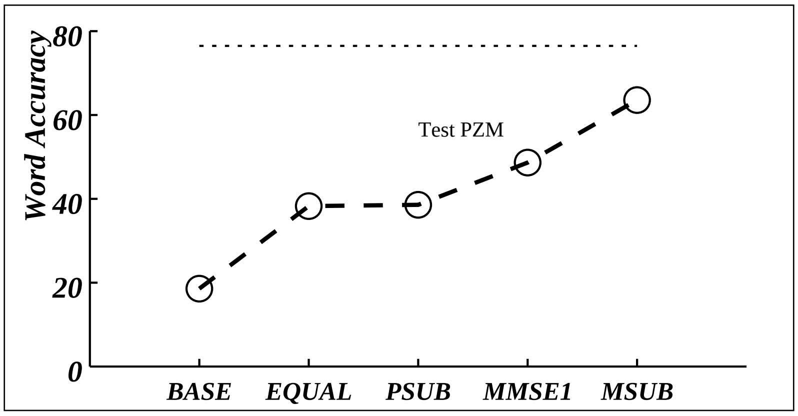



**Figure 8-1:** Independent compensation for noise and filtering in the spectral domain. Comparison of the baseline performance (BASE), cepstral equalization (EQUAL), Power Spectral Subtraction (PSUB), Minimum Mean Squared Error with one curve for all frequencies (MMSE1) and Magnitude Spectral Subtraction (MSUB) when trained on the CLSTK microphone and tested on the CRPZM. The broken line represents the word accuracy of the system trained and tested on the CRPZM.

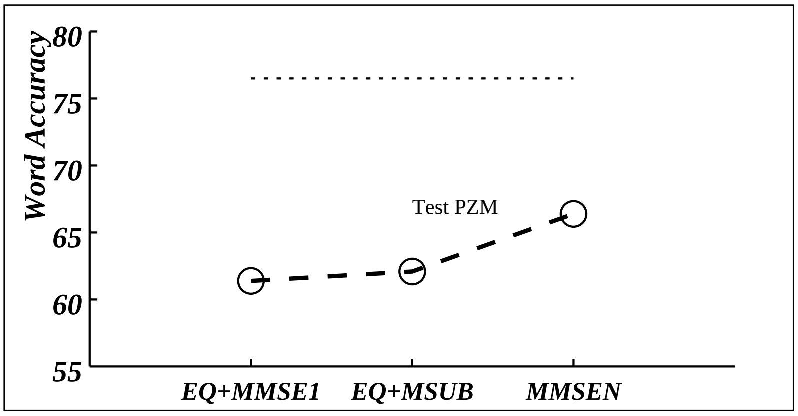



**Figure 8-2:** Comparison of algorithms that perform independent compensation for both noise and filtering (EQ+MMSE1, EQ+MSUB) versus an algorithm that performs joint compensation for noise and filtering (MMSEN) in the spectral domain. The Figure shows the word accuracy of SPHINX when trained on the CLSTK microphone and tested on the CRPZM. The broken line represents the word accuracy of the system trained and tested on the CRPZM.

CRPZM recordings. Also, the performance of the MMSEN algorithm is higher than that of the cascade of equalization EQUAL and noise suppression MMSE1, with this difference being statistically significant.

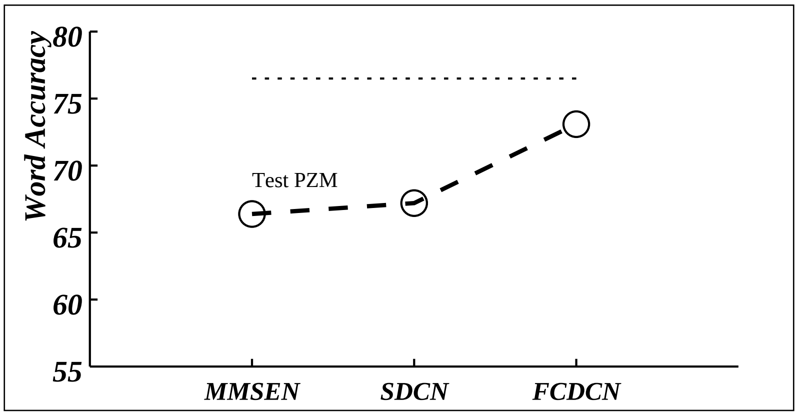



**Figure 8-3:** Comparison of the performance of algorithms in the spectral domain (MMSEN) and the cepstral domain (SDCN,FCDCN). All these algorithms perform joint compensation for noise and filtering but they also require stereo data. MMSEN uses a minimum mean squared error criterion with one transformation curve per frequency component, and it is described in Chapter 4. SDCN, SNR-Dependent Cepstral Normalization , and the FCDCN, Fixed Codeword-Dependent Cepstral Normalization are described in Chapters 4 and 6 respectively. SPHINX was trained on the CLSTK microphone and tested on the CRPZM. The broken line represents the word accuracy of the system trained and tested on the CRPZM.

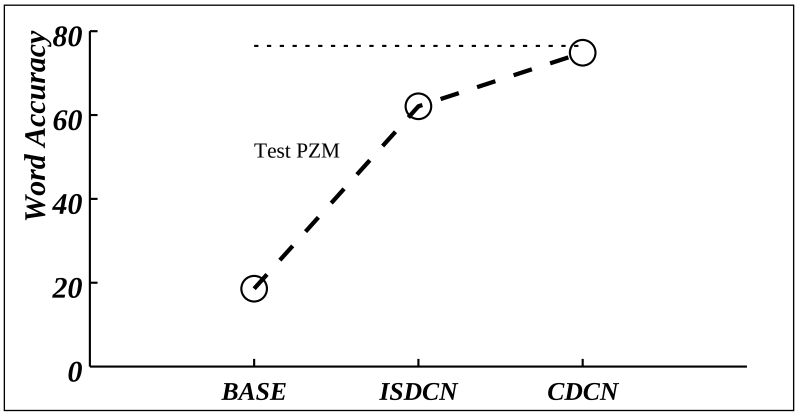



**Figure 8-4:** Comparison of algorithms that adapt to new acoustical environments (ISDCN and CDCN) algorithms when trained on the CLSTK microphone and tested on the CRPZM. BASE represents no processing, ISDCN is the Interpolated SNR-Dependent Cepstral Normalization described in Chapter 6 and CDCN is the Codeword-Dependent Cepstral Normalization described in Chapter 5. ISDCN and CDCN perform also joint compensation for noise and filtering. The broken line represents the word accuracy of the system trained and tested on the CRPZM.

Figure 8-3 shows a comparison of our best algorithm operating in the frequency domain (MMSEN) and the algorithms derived in the cepstral domain (SDCN, FCDCN). All three algorithms perform joint compensation for noise and filtering. The SDCN algorithm performs at the same level than the MMSEN algorithm but it is simpler computationally, as it only requires to compensate the first two cepstral components. The FCDCN on the other hand performs substantially better than both the MMSEN and the SDCN algorithms as the compensation vector is *codeword-dependent*.

Figure 8-4 shows the performance of ISDCN and CDCN as algorithms that adapt to new acoustical environments. Both of them operate in the cepstral domain and perform joint compensation for noise and filtering. As can be seen, use of the CDCN algorithm brings the accuracy of the system when trained on the CLSTK and tested on the CRPZM to the level obtained when the system is trained and tested on the CRPZM.

# 9 Conclusions

In this dissertation we have addressed the problem of building speech recognition systems that are robust to changes in the acoustical environment. With the development of large-vocabulary continuous-speech speaker-independent recognition systems like SPHINX, three of the major problems in speech recognition have been solved. However, for unconstrained speech recognition we need to build systems that will work for ungrammatical spontaneous speech, speakers with different dialects, and in real acoustical environments.

Not only have we learned many lessons through the course of this dissertation about the problems in environment independence, but we have also provided some solutions that can be used in practice.

## 9.1. Contributions

The major contribution of this work is to show that an increased robustness to changes in the environment can be achieved. Use of our algorithms on a system trained on clean speech brings the performance when testing on a given environment to at least the level obtained when the system was trained and tested on that particular environment, and this can be accomplished without the need for retraining on data from the new environment. When a signal is corrupted by noise some information is inevitably lost. While we have not overcome the degradation present in noisy speech in our work, we believe that the algorithms described provide a step in the right direction.

We have proposed specific algorithms that adapt to new acoustical environments without the need for retraining. The CDCN, *Codeword-Dependent Cepstral Normalization* estimates the noise and equalization cepstral vectors that maximizes the probability of an ensemble of input cepstral frames. The ISDCN, *Interpolated SNR-Dependent Cepstral Normalization* estimates the environmental parameters by minimizing the accumulated distortion in the vector quantizer. The recognition accuracy when testing on speech recorded with a desk-mounted microphone on a system trained on speech recorded with a headset-mounted microphone is essentially the same obtained when the system is trained and tested with the recordings from the desk-mounted microphone.

We have shown that *joint compensation* for noise and spectral equalization is more effective than a combination of independent compensators. The correction vectors are an interpolating function between the equalization vector at high SNR and the noise vector at low SNR. Conditioning on the *instantaneous SNR* can provide processing benefits while retaining simplicity. We have identified instantaneous SNR as the primary variable in the normalization procedure.

The use of a *universal codebook* is the main tool that allows us to track deviations from a standard acoustical ambience by finding the parameters of the transformation that best "matches" different acoustic spaces. Therefore, a universal codebook that is trained once and for all can be used instead of training different codebooks for different applications. Vocabulary-independent models (Hon and Lee [35]) will benefit from this approach, as those models need to be environment-independent as well.

We have shown that the *cepstral domain* is a viable parameter domain for noise suppression. The advantage of performing normalization in the cepstral domain is that we can integrate it better with the rest of a system like SPHINX that uses cepstral parameters as feature vectors.

The use of the *EM algorithm* as a tool for obtaining maximum likelihood estimates has been essential to the robust estimation of parameters used in this dissertation.

We have shown that ISDCN and FCDCN can be implemented by modifying the VQ stage, resulting in an algorithm that is very *efficient* computationally.

We have introduced the use of *stereo databases* for evaluation purposes. While it is difficult to compare the performance of algorithms when there is not a standard database that all researchers use, some criteria can be established that will help to make comparisons. We proposed the use of two reference figures: the error rate when the system is trained and tested on clean speech and the error rate when the system is trained and tested on the noisy speech. These references will serve as indicators of the performance of our algorithms when the system is trained on clean speech and tested on noisy speech.

We have also introduced the use of *speech and noise spectral averages* as a more informative feature than just SNR as a characterization of an environment. When speech can be passed through a linear filter and the noise is not white, SNR measurements can provide a misleading indication of how a speech recognizer will perform if no normalization is accomplished.

We have proposed a method for *frequency normalization* via variable warping of the frequency axis. Although this is something that many researchers have attempted, it is not an easy task because the frequency alignment is very unconstrained. The use of the bilinear transform with only one free parameter has proven to be effective in normalizing the the nominal resonance frequencies of different speakers.

## 9.2. Suggestions for Future Work

Even though we have taken a big step forward in being able to transfer speech recognition systems from the laboratory to the real world, the field of environment-independent recognition is still in its infancy. In this section we describe some future research that appears to be worthwhile.

One of the problems with the ISDCN and CDCN algorithms is that obtaining estimates of the equalization vector via maximum likelihood does not use any *a priori* information that may be available about the environment. Although the estimates are unbiased, they may exhibit a large variance if not enough samples are used. We have observed that in some cases, these estimates may not represent legitimate equalization vectors. A topic to investigate in future research is the use of *a priori* information on the distribution of the equalization vector. One possibility would be to include a small codebook of different environments. The highly accurate FCDCN algorithm could be used with different sets of correction vectors depending on which environment is selected. As always, the criterion could still be minimization of the VQ distortion.

The use of a different criterion for the parameter selection should also be considered in future research. In this dissertation we have used the minimization of the VQ distortion as the criterion to estimate all the vectors. It may well be the case, however, that a criterion such as maximization of the probability of the utterance or minimization of the probability of error could lead to better performance. Although minimization of the probability of error would be the goal of any speech recognition system, it would be extremely difficult to use as the basis for a distortion metric. Maximization of the probability of the utterance, however, is a plausible alternative and it is the one used by HMMs. In other words, we could leave the decision of environmental parameters to the HMM search, so that we are not penalized by early decisions. Although there could be a small improvement in performance if this is done at the VQ stage by using tied mixtures or semi-continuous HMM models (Huang *et al.* [36]), we believe that most of the benefit would come from a more accurate estimation of the environmental parameters.

We have found that all algorithms described in this dissertation can potentially perform rather well for high SNRs, and that the problems arise when frames with low SNR are processed. For those frames, the algorithms select just one "cleaned" vector whereas there will be several possible that have been masked by the noise. Selecting one of them and not including the other alternatives is a serious flaw of the present approaches, as it would be better to let the HMM search makes its decision with all information available. One way of doing this would be rather than using discrete models to use semi-continuous ones in which the relative probabilities of different codewords are passed to the HMM search. Another possible alternative would be to use some sort *phone-dependent* rather than *codeword-dependent* correction vectors. Specifically, we propose the use of different sets of correction vectors per phone (using perhaps about 40 phones). Many different correction vectors and VQ labels would be obtained for every frame of speech depending on what phone was hypothesized. The HMM search would select the most likely phone string.

A longer time frame can help in the compensation process. In this dissertation, compensation was based only on knowledge of the current frame, while it is clear that using information on the adjacent frames will benefit the accuracy of the estimate. A possible extension of the techniques developed here would be to incorporate the same processing to differential parameters.

The algorithms described in this dissertation could also be used for speech enhancement. The CDCN algorithm could be used to obtain a better spectral estimate of the clean speech than that of conventional noise suppression algorithms. Wiener filtering could then be used to enhance the signal.

More work remains to be done on the problem of real-time implementation. The algorithms described in this dissertation aim to normalize long-term characteristics of the speech signal in addition to the short-term ones used by most systems. Allowing for a slowly changing environment requires more research on the rate of change and the mechanisms for updating. In this work we have shown that using the previous utterance as a unit is a reasonable choice, but that the use of longer estimation times will yield more accurate estimates.

This dissertation has not explicitly addressed the problem of interference by additive non-stationary noise. We are optimistic that the technique of noise-word modeling described by (Ward [81]) will provide some additional improvement for speech collected in the presence of non-stationary interference sources such as slamming doors, ringing telephones, etc. This use of noise-word models is a complementary technique to the algorithms described in this dissertation.

It is necessary to investigate the behavior of many different microphones and environments so that more general conclusions can be drawn. It would also be desirable to test these algorithms with telephone speech.

The absence of a standard database for evaluation of algorithms is definitely not benefiting the field of environment-independent recognition. The continuous speech recognition community has benefited from the existence of common tasks and databases so that direct comparison between algorithms is more straightforward. Our hope is that in the near future similar efforts can be directed toward the development of standard databases for environment-independent recognition.

# Appendix A
# Glossary

## A.1. Time Domain

| | |
|---|---|
| $x[m]$ | The clean signal |
| $y[m]$ | The noisy signal |
| $h[m]$ | The filter's impulse response |
| $n[m]$ | The noise process |

## A.2. Frequency Domain

| | |
|---|---|
| $X_i(\omega)$ | Power Spectral Density of clean signal at frame i |
| $Y_i(\omega)$ | Power Spectral Density of noisy signal at frame i |
| $N(\omega)$ | Power Spectral Density of the noise |
| $Q(\omega) = \lvert H(\omega) \rvert^2$ | Magnitude squared of the filter's frequency response |
| $\mathbf{X}_i(\omega) = \ln X_i(\omega)$ | Log-spectrum of the clean signal at frame *i* |
| $\mathbf{Y}_i(\omega) = \ln Y_i(\omega)$ | Log-spectrum of the noisy at frame *i* |
| $\mathbf{N}(\omega) = \ln N(\omega)$ | Log-spectrum of the noise |
| $\mathbf{Q}(\omega) = \ln Q(\omega)$ | Equalization transfer function |
| $\overline{\mathbf{X}}_i(\omega) = \mathbf{X}_i(\omega) - \mathbf{N}_i(\omega)$ | Normalized log-spectrum of the clean signal at frame *i* |
| $\overline{\mathbf{Y}}_i(\omega) = \mathbf{Y}_i(\omega) - \mathbf{N}_i(\omega)$ | Normalized log-spectrum of the noisy signal at frame *i* |

## A.3. Cepstral Domain

| | |
|---|---|
| $\mathbf{x}_i$ | Cepstral vector of the clean signal at frame *i* |
| $\mathbf{y}_i$ | Cepstral vector of the noisy signal at frame *i* |
| $\mathbf{z}_i$ | Observed cepstral vector at frame *i* |
| $Z$ | Collection of cepstral vectors for an utterance |
| $\mathbf{n}$ | Cepstral vector of the noise |
| $\mathbf{q}$ | Cepstral equalization vector |
| $\mathbf{r}[\mathrm{k}]$ | Correction vector **r** for the $k^{th}$ codeword |
| $\mathbf{s}[\mathrm{k}]$ | Correction vector **s** for the $k^{th}$ codeword |
| $\mathbf{w}(\mathrm{SNR})$ | Correction vector **w** as a function of SNR |
| $\mathbf{c}[\mathrm{k}]$ | Vector of the $k^{th}$ codeword |

## A.4. Indices

The following indices refer to

| | |
|---|---|
| $i$ | Frames in an utterance |
| $k$ | Codewords in codebook |
| $j$ | Iteration number in iterative algorithms |

## A.5. Probabilistic Models

| | |
|---|---|
| $S$ | The word string |
| $\nu$ | The noise model |
| $\xi$ | The speech model |
| $P[k]$ | *A priori* probability for $k^{th}$ mixture component |
| $\Sigma_k$ | Covariance matrix of $k^{th}$ mixture component |

# Appendix B
# Signal Processing in SPHINX

The speech signal $x[m]$ is digitized at a sampling rate of 16 kHz and multiplied by a Hamming window $h[m]$ of $N = 320$ samples (20 ms) every $M = 160$ samples (10 ms).

$$h[m] = 0.54 - 0.46\cos\left(\frac{2\pi m}{N-1}\right) \quad 0 \le m \le N-1 \tag{B.1}$$

$$x_i[m] = x[iM + m]\,h[m] \quad 0 \le m \le N-1 \tag{B.2}$$

where $x_i[m]$ represents frame $i$. After this, a high-pass preemphasis filter is applied

$$y_i[m] = x_i[m] - 0.97\,x_i[m-1] \tag{B.3}$$

The preemphasized windowed signal $y_i[m]$ is used to compute $p = 14$ autocorrelation coefficients

$$R[k] = \sum_{m=0}^{N-1-k} y_i[m]\,y_i[m+k] \quad 0 \le k \le p \tag{B.4}$$

that are multiplied by a pascal lag window with $\tau = 1500$

$$v[k] = \frac{\binom{\tau-k+1}{k}}{\binom{\tau+k-1}{k}} \tag{B.5}$$

that has the form in Figure B-1.

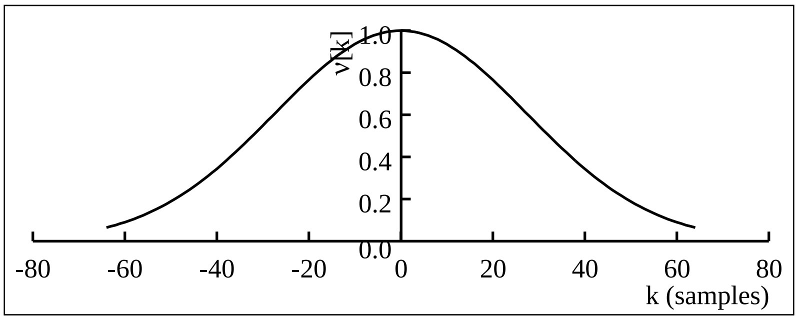



**Figure B-1:** Pascal window for τ = 1500

The lag window will suppress the harmonics that will appear at multiples of the pitch period, which can be as low as 32 samples at 16 kHz for a high pitched female voice with a fundamental frequency of 500 Hz. The Levinson-Durbin recursion (Rabiner and Schafer [68]) is used to obtain the LPC parameters from the windowed autocorrelation coefficients

$$E^{(0)} = R[0] \tag{B.6}$$

$$k_i = \frac{R[i] - \sum_{j=1}^{i-1} a_j^{(i-1)} R[i-j]}{E^{(i-1)}} \quad 1 \le i \le p \tag{B.7}$$

$$a_i^{(i)} = k_i \tag{B.8}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad 1 \le j \le i-1 \tag{B.9}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{B.10}$$

Equations (B.7) - (B.10) are solved recursively for $i = 1, 2, \ldots, p$ with the final solution being given by

$$a_j = a_j^{(p)} \quad 1 \le j \le p \tag{B.11}$$

The LPC parameters define an all-pole system

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{B.12}$$

whose logarithm can be expressed as a Fourier series with the *complex* cepstrum. Since it is obtained from an LPC analysis, this is called the LPC-cepstrum.

$$\ln H(z) = \sum_{n=0}^{\infty} \mathbf{c}[n] z^{-n} \tag{B.13}$$

Atal [2] showed that taking the logarithm of (B.12), equating it to (B.13) and taking derivatives with respect to $z^{-1}$ leads to the following recursion for $c[n]$

$$\mathbf{c}[0] = \ln(G)$$

$$\mathbf{c}[n] = a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) c[k]\, a_{n-k} \quad 1 \le n \tag{B.14}$$

Even though there is an infinite number of cepstrum coefficients, we truncated the sequence to 32. Finally a bilinear transform with warping parameter $\alpha = 0.6$ is applied as described in Appendix C, obtaining 12 frequency-warped LPC-cepstral coefficients.

# Appendix C
# The Bilinear Transform

The transformation defined by

$$z_{new}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{C.1}$$

for $0 < \alpha < 1$ belongs to the class of so-called bilinear transforms. It is a mapping in the complex plane that maps the unit circle onto itself. The frequency transformation is obtained by making the substitution $z = e^{j\omega}$ and $z_{new} = e^{j\omega_{new}}$:

$$\omega_{new} = \omega + 2\,\mathrm{arctg}\left[\frac{\alpha\sin(\omega)}{1 - \alpha\cos(\omega)}\right] \tag{C.2}$$

An algorithm to perform this frequency warping transformation on a time signal is described by Oppenheim and Johnson [61]. If we assume a causal finite time sequence $c[n]$, with $c[n] = 0$ for $n < 0$ and $n > p$, the warped sequence $c_w[n]$ will contain an *infinite* number of coefficients that can be obtained by passing the reversed input sequence through a cascade of filters and sampling their outputs at time $n = 0$ as shown in Figure C-1.

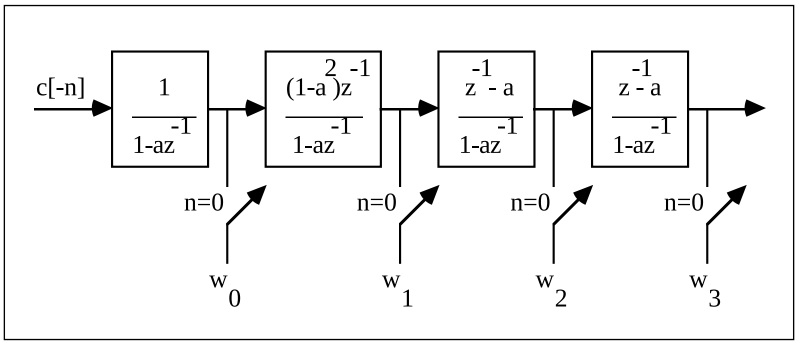



**Figure C-1:** Bilinear transform algorithm as a linear filtering operation. By having as the input to this stage of filters a time-reversed cepstrum sequence, we can obtain the corresponding warped coefficients as the outputs of these filters at time $n = 0$.

This algorithm is used by SPHINX to do the warping transformation on the cepstral coefficients. We have to use the *complex* cepstrum rather than the *real* cepstrum[28] because the sequence $c[n]$ has to be causal to use Oppenheim's algorithm.

We have noticed that if only p coefficients are retained for $c_w[n]$, we can relate the input/output relationship by a linear transformation of the form

$$\mathbf{c}_w = \mathbf{L}(\alpha)\,\mathbf{c} \tag{C.3}$$

where $\mathbf{c}$ and $\mathbf{c}_w$ are the input and output vectors of length $p$ and $\mathbf{L}$ is the warping matrix. Nocerino *et al.* [59] also pointed out that a general warping transformation is equivalent to a matrix multiplication. We see that every coefficient in the warped sequence $c_w[n]$ is a linear combination of the coefficients of the original sequence $c[n]$. If the warping parameter $\alpha$ is small, it can be shown that neglecting all powers of $\alpha$ greater than 1, the finite length sequence $\{c[n], n=0,1,\ldots,p\}$ is transformed as

$$c_w[n] = -(n-1)\,\alpha\, c[n-1] + c[n] + (n+1)\,\alpha\, c[n+1] \tag{C.4}$$

## C.1. Cascade of Warping Stages

We now show that the application of two stages of the bilinear transform with warping parameters $\alpha_1$ and $\alpha_2$ is equivalent to applying one stage of the bilinear transform with $\alpha = (\alpha_1 + \alpha_2 / (1 + \alpha_1\alpha_2)$.

Let $z$ be the complex variable in the original and $s$ and $u$ the complex variable after one and two bilinear transforms. The relationship between them is

$$s^{-1} = \frac{z^{-1} - \alpha_1}{1 - \alpha_1 z^{-1}} \tag{C.5}$$

$$u^{-1} = \frac{s^{-1} - \alpha_2}{1 - \alpha_2 s^{-1}} \tag{C.6}$$

Combining (C.5) and (C.6) we obtain

$$u^{-1} = \frac{\dfrac{z^{-1} - \alpha_1}{1 - \alpha_1 z^{-1}} - \alpha_2}{1 - \alpha_2\,\dfrac{z^{-1} - \alpha_1}{1 - \alpha_1 z^{-1}}} = \frac{z^{-1}(1 + \alpha_1\alpha_2) - (\alpha_1 + \alpha_2)}{(1 + \alpha_1\alpha_2) - (\alpha_1 + \alpha_2)z^{-1}} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{C.7}$$

where the new $\alpha$ can be expressed as

$$\alpha = \frac{\alpha_1 + \alpha_2}{1 + \alpha_1\alpha_2} \tag{C.8}$$

---

[28]The *real* cepstrum is the even part of the *complex* cepstrum. The interested reader is referred to Rabiner and Schafer [68] for details.

# Appendix D
# Spectral Estimation Issues

In this appendix we discuss the characterization of the LPC-cepstrum spectral estimator as a zero-mean gaussian random vector. The use of a finite data sample and possible inadequacies of the model will not make possible to obtain the exact PSD (Power Spectral Density) of the speech signal.

The PSD of a random process cannot be inferred from a sample function of the process. We know that if the random process is ergodic, an infinite sample function has all the information required to obtain the PSD of that process. Since speech is a non-stationary random process, the spectral estimation techniques used are based on quasi-stationarity or short-term analysis. The data window used in SPHINX is 320 points, which is short enough as to assume stationarity and long enough as to yield reasonable estimates.

The statistics of the LPC parameters and the AR Power Spectral Density Estimator are not available even for an analysis of order 1. Kay [43] showed that the spectrum of the LPC estimator is asymptotically gaussian (when both the number of points and the analysis order tend to infinity). Furthermore, the log-spectral estimate has a variance independent of the mean value.

By using an LPC analysis, we have made the assumption that the speech signal can be characterized as an AR process. Since for instance, nasals present zeroes as well as poles, there will be some inadequacies by using the LPC-cepstrum.

We assume that the frequency-warped LPC-cepstrum $\mathbf{z}$ computed by the SPHINX front-end is a noisy estimate of the true *Power Cepstral Density* $\mathbf{y}$. In the absence of exact statistics we modeled the *pdf* $p(\mathbf{z}/\mathbf{y})$ as a gaussian random vector $N_z(\mathbf{y}, \Gamma)$. We have confirmed the validity of the gaussian assumption empirically for the frequency-warped LPC-cepstrum in SPHINX.

# Appendix E
# MMSE Estimation in the CDCN Algorithm

In this section we derive the expressions for the conditional MMSE estimate $E[\mathbf{x}|\mathbf{z},\mathbf{n},\mathbf{q}]$, as well as a gaussian decomposition for the densities involved.

## E.1. The Conditional Probability in the CDCN Algorithm

Let's express the conditional probability of the clean vector **x** given the observation **z** and the environmental parameters **n** and **q** as a function of the mixture $k$:

$$p(\mathbf{x}|\mathbf{z},\mathbf{n},\mathbf{q}) = \sum_{k=0}^{K-1} p(\mathbf{x},k|\mathbf{z},\mathbf{n},\mathbf{q}) \tag{E.1}$$

where each term in the sum is according to Bayes Rule

$$p(\mathbf{x},k|\mathbf{z},\mathbf{n},\mathbf{q}) = \frac{p(\mathbf{z},\mathbf{x},k|\mathbf{n},\mathbf{q})}{p(\mathbf{z}|\mathbf{n},\mathbf{q})} \tag{E.2}$$

whose numerator can be expressed as

$$p(\mathbf{z},\mathbf{x},k|\mathbf{n},\mathbf{q}) = p(\mathbf{z}|\mathbf{x},k,\mathbf{n},\mathbf{q})\, p(\mathbf{x},k|\mathbf{n},\mathbf{q}) \tag{E.3}$$

Since the distribution of **x** does not depend on the noise nor the filter

$$p(\mathbf{x},k|\mathbf{n},\mathbf{q}) = p(\mathbf{x},k) = P[k]\, p(\mathbf{x}|k) \tag{E.4}$$

where $p(\mathbf{x}|k)$ is the mixture $k$ and $P[k]$ is the *a priori* probability for mixture $k$. Combining (E.3) and (E.4), the numerator in (E.2) can be expressed as

$$p(\mathbf{z},\mathbf{x},k|\mathbf{n},\mathbf{q}) = P[k]\, p(\mathbf{z}|\mathbf{x},k,\mathbf{n},\mathbf{q})\, p(\mathbf{x}|k) \tag{E.5}$$

with the denominator in (E.2) having the form

$$p(\mathbf{z}|\mathbf{n},\mathbf{q}) = \sum_{k=0}^{K-1} \int p(\mathbf{z},\mathbf{x},k|\mathbf{n},\mathbf{q})\, d\mathbf{x} \tag{E.6}$$

Combining Equations (E.1), (E.2), (E.5) and (E.6) we obtain the following expression for the *a posteriori* probability:

$$p(\mathbf{x}|\mathbf{z},\mathbf{n},\mathbf{q}) = \frac{\sum_{k=0}^{K-1} P[k]\, p(\mathbf{z}|\mathbf{x},\mathbf{n},\mathbf{q},k)\, p(\mathbf{x}|k)\, d\mathbf{x}}{\sum_{k=0}^{K-1} P[k] \int p(\mathbf{z}|\mathbf{x},\mathbf{n},\mathbf{q},k)\, p(\mathbf{x}|k)\, d\mathbf{x}} \tag{E.7}$$

With the *a posteriori* probability (E.7) we can derive the MMSE estimate for $\mathbf{x}$ as:

$$\hat{\mathbf{x}}_{MMSE} = E[\mathbf{x} \,|\, \mathbf{z}, \mathbf{n}, \mathbf{q}] = \frac{\sum_{k=0}^{K-1} P[k] \int \mathbf{x}\, p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)\, d\mathbf{x}}{\sum_{k=0}^{K-1} P[k] \int p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k)\, d\mathbf{x}} \tag{E.8}$$

where $p(\mathbf{x}|k)$ is the $k^{th}$ mixture component and $p(\mathbf{z}|\mathbf{x}, k, \mathbf{n}, \mathbf{q})$ is the the *pdf* of the spectral estimator $p(\mathbf{z}|\mathbf{y})$, both assumed to be gaussian random vectors.

## E.2. Gaussian Decomposition

In this section we derive a decomposition for the product of gaussians

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}, \mathbf{n}, \mathbf{q}, k)\, p(\mathbf{x}|k) &= \frac{1}{\alpha|\Sigma_k|^{1/2}} exp[-\frac{1}{2}(\mathbf{x} - \mathbf{c}[k])^T \Sigma_k^{-1} (\mathbf{x} - \mathbf{c}[k])] \\ &+ \frac{1}{\alpha|\Gamma|^{1/2}} exp[-\frac{1}{2}(\mathbf{z} - \mathbf{x} - \mathbf{q} - \mathbf{r}(\mathbf{x}))^T \Gamma^{-1} (\mathbf{z} - \mathbf{x} - \mathbf{q} - \mathbf{r}(\mathbf{x}))] \end{aligned} \tag{E.9}$$

that is more convenient for integration of (E.8). By grouping the terms that depend on $\mathbf{x}$ in the exponent of (E.9) we obtain an alternate expression in terms of vector $\mathbf{b}(\mathbf{x}, k)$ and the scalar $d(\mathbf{x}, k)$

$$\begin{aligned} &(\mathbf{z} - \mathbf{x} - \mathbf{q} - \mathbf{r}(\mathbf{x}))^T \Gamma^{-1} (\mathbf{z} - \mathbf{x} - \mathbf{q} - \mathbf{r}(\mathbf{x})) + (\mathbf{x} - \mathbf{c}[k])^T \Sigma_k^{-1} (\mathbf{x} - \mathbf{c}[k]) \\ &= (\mathbf{x} - \mathbf{b}(\mathbf{x}, k))^T (\Gamma^{-1} + \Sigma_k^{-1}) (\mathbf{x} - \mathbf{b}(\mathbf{x}, k)) + d(\mathbf{x}, k) \end{aligned} \tag{E.10}$$

By equating the terms in $\mathbf{x}$ in (E.10) we obtain

$$\Gamma^{-1} (\mathbf{z} - \mathbf{q} - \mathbf{r}(\mathbf{x})) + \Sigma_k^{-1} \mathbf{c}[k] = (\Gamma^{-1} + \Sigma_k^{-1})\, \mathbf{b}(\mathbf{x}, k) \tag{E.11}$$

Also equating the zero order terms in (E.10), we get an expression for $d(\mathbf{x}, k)$ of the form

$$\begin{aligned} d(\mathbf{x}, k) &= (\mathbf{z} - \mathbf{q} - \mathbf{r}(\mathbf{x}))^T \Gamma^{-1} (\mathbf{z} - \mathbf{q} - \mathbf{r}(\mathbf{x})) \\ &+ \mathbf{c}[k] \Sigma_k^{-1} \mathbf{c}[k] - \mathbf{b}(\mathbf{x}, k)^T (\Gamma^{-1} + \Sigma_k^{-1})\, \mathbf{b}(\mathbf{x}, k) \end{aligned} \tag{E.12}$$

We will now try to obtain closed-form expressions for $\mathbf{b}(\mathbf{x}, k)$ and $d(\mathbf{x}, k)$ by using the following matrix identities:

$$\begin{aligned} &(\Gamma^{-1} + \Sigma_k^{-1})^{-1} \Gamma^{-1} = (\Gamma(\Gamma^{-1} + \Sigma_k^{-1}))^{-1} \\ &= (I + \Gamma \Sigma_k^{-1})^{-1} = ((\Sigma_k + \Gamma) \Sigma_k^{-1})^{-1} = \Sigma_k (\Sigma_k + \Gamma)^{-1} \end{aligned} \tag{E.13}$$

and similarly

$$(\Gamma^{-1} + \Sigma_k^{-1})^{-1} \Sigma_k^{-1} = \Gamma (\Sigma_k + \Gamma)^{-1} \tag{E.14}$$

We get an expression for $\mathbf{b}(\mathbf{x}, k)$ by combining (E.11), (E.13) and (E.14):

$$\mathbf{b}(\mathbf{x}, k) = \Sigma_k (\Sigma_k + \Gamma)^{-1} (\mathbf{z} - \mathbf{q} - \mathbf{r}(\mathbf{x})) + \Gamma (\Sigma_k + \Gamma)^{-1} \mathbf{c}[k] \tag{E.15}$$

Combining (E.11) and (E.15), we obtain

$$\mathbf{b}(\mathbf{x},k)^T(\Gamma^{-1}+\Sigma_k^{-1})\,\mathbf{b}(\mathbf{x},k) = (\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x}))^T\Gamma^{-1}\Sigma_k(\Gamma+\Sigma_k)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x}))$$

$$+\,\mathbf{c}^T[k]\,\Sigma_k^{-1}\,\Gamma(\Sigma_k+\Gamma)^{-1}\mathbf{c}[k] + 2\,\mathbf{c}^T[k]\,(\Sigma_k+\Gamma)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})) \qquad (E.16)$$

By using the following matrix identities we obtain:

$$\Gamma^{-1} - \Gamma^{-1}\Sigma_k(\Gamma+\Sigma_k)^{-1} = \Gamma^{-1}(I - \Sigma_k(\Sigma_k+\Gamma)^{-1})$$
$$= \Gamma^{-1}((\Sigma_k+\Gamma)-\Sigma_k)(\Sigma_k+\Gamma)^{-1} = (\Sigma_k+\Gamma)^{-1} \qquad (E.17)$$

and similarly

$$\Sigma_k^{-1} - \Sigma_k^{-1}\Gamma(\Gamma+\Sigma_k)^{-1} = (\Sigma_k+\Gamma)^{-1} \qquad (E.18)$$

The solution for $d(\mathbf{x},k)$ using (E.12), (E.16), (E.17) and (E.18) is:

$$d(\mathbf{x},k) = (\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})-\mathbf{c}[k])^T(\Gamma+\Sigma_k)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}(\mathbf{x})-\mathbf{c}[k]) \qquad (E.19)$$

The relationship between the determinants of the covariance matrices is:

$$|\Gamma^{-1}+\Sigma_k^{-1}| = |\Gamma^{-1}(\Gamma+\Sigma_k)\Sigma_k^{-1}| = |\Gamma|^{-1}|\Gamma+\Sigma_k||\Sigma_k|^{-1} \qquad (E.20)$$

or alternatively

$$|\Gamma|^{1/2}|\Sigma_k|^{1/2} = |(\Gamma^{-1}+\Sigma_k^{-1})^{-1}|^{1/2}|\Gamma+\Sigma_k|^{1/2} \qquad (E.21)$$

so, by using (E.10), (E.15), (E.19) and (E.21), the product of gaussians in (E.8) has the form:

$$p(\mathbf{z}|\mathbf{x},\mathbf{n},\mathbf{q},k)\,p(\mathbf{x}|k) = N_{\mathbf{x}}(\mathbf{b}(\mathbf{x},k),(\Gamma^{-1}+\Sigma_k^{-1})^{-1})\,N_{\mathbf{z}}(\mathbf{q}+\mathbf{r}(\mathbf{x})+\mathbf{c}[k],\Gamma+\Sigma_k) \qquad (E.22)$$

and this concludes our decomposition. Integrating (E.22) with respect to $\mathbf{x}$ is not possible since $\mathbf{b}(\mathbf{x},k)$ is a function of $\mathbf{x}$ through $\mathbf{r}(\mathbf{x})$. If we make the approximation that $\mathbf{r}(\mathbf{x}) = \mathbf{r}(\mathbf{c}_k) = \mathbf{r}[k]$ is constant for every mixture we can obtain easily

$$\int p(\mathbf{z}|\mathbf{x},\mathbf{n},\mathbf{q},k)\,p(\mathbf{x}|k)\,d\mathbf{x} = N_{\mathbf{z}}(\mathbf{q}+\mathbf{r}[k]+\mathbf{c}[k],\Gamma+\Sigma_k) \qquad (E.23)$$

$$\int \mathbf{x}\,p(\mathbf{z}|\mathbf{x},\mathbf{n},\mathbf{q},k)\,p(\mathbf{x}|k)\,d\mathbf{x} = \mathbf{b}[k]\,N_{\mathbf{z}}(\mathbf{q}+\mathbf{r}[k]+\mathbf{c}[k],\Gamma+\Sigma_k) \qquad (E.24)$$

with $\mathbf{b}[k]$ being given by

$$\mathbf{b}[k] = \Sigma_k(\Sigma_k+\Gamma)^{-1}(\mathbf{z}-\mathbf{q}-\mathbf{r}[k]) + \Gamma(\Sigma_k+\Gamma)^{-1}\mathbf{c}[k] \qquad (E.25)$$

Equations (E.23), (E.24) and (E.25) will be used in the derivation of the CDCN algorithm.

# Appendix F
# Maximum Likelihood via the EM Algorithm

The EM (Estimate-Maximize) algorithm is a general method to solve maximum likelihood problems with incomplete data. It was developed by Laird *et al.* [44]. First we present the Jensen's inequality that is used in deriving the EM algorithm.

## F.1. Jensen's Inequality

Jensen's inequality deals with expectations of convex functions. In this subsection we will derive it for the specific case of the logarithm which is of interest for the EM algorithm.

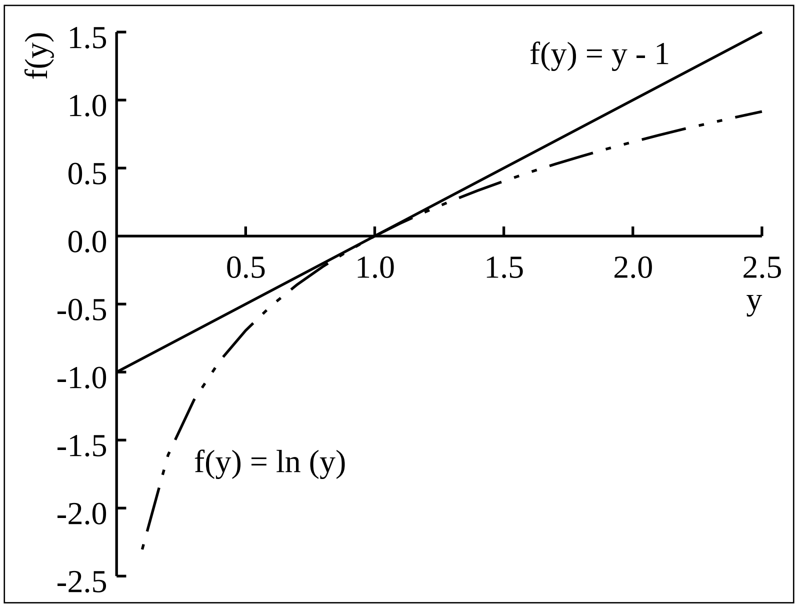



**Figure F-1:** Functions $f(y) = y - 1$ and $f(y) = \ln(y)$.

It is clear from inspection of Figure F-1 that

$$y - 1 \geq \ln y, \quad \forall y \qquad (F.1)$$

Let $f_{\mathbf{X}}(\mathbf{x})$ and $g_{\mathbf{X}}(\mathbf{x})$ be two legitimate pdfs so that

$$\int f_{\mathrm{X}}(\mathbf{x})\, d\mathbf{x} = \int g_{\mathrm{X}}(\mathbf{x})\, d\mathbf{x} = 1 \tag{F.2}$$

or after some manipulation

$$0 = \int [\, f_{\mathrm{X}}(\mathbf{x}) - g_{\mathrm{X}}(\mathbf{x})]\, d\mathbf{x} = \int f_{\mathrm{X}}(\mathbf{x}) \left[ \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} - 1 \right] d\mathbf{x} \tag{F.3}$$

for all vectors **x** whose $f_{\mathrm{X}}(\mathbf{x}) \neq 0$. For those vectors we have that

$$y = \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} \geq 0 \tag{F.4}$$

since both $g_{\mathrm{X}}(\mathbf{x}) \geq 0$ and $f_{\mathrm{X}}(\mathbf{x}) \geq 0$ for two legitimate pdfs. Substituting (F.4) in (F.1) we obtain

$$\frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} - 1 \geq \ln \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} \tag{F.5}$$

Since $f_{\mathrm{X}}(\mathbf{x}) \geq 0$, multiplying both sides of (F.5) by $f_{\mathrm{X}}(\mathbf{x})$ maintains the inequality sign, yielding:

$$f_{\mathrm{X}}(\mathbf{x}) \left[ \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} - 1 \right] \geq f_{\mathrm{X}}(\mathbf{x}) \ln \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} \tag{F.6}$$

that after integration with respect to **x** combined with (F.3) gives:

$$0 \geq \int f_{\mathrm{X}}(\mathbf{x}) \ln \frac{g_{\mathrm{X}}(\mathbf{x})}{f_{\mathrm{X}}(\mathbf{x})} d\mathbf{x} \tag{F.7}$$

or finally:

$$\int f_{\mathrm{X}}(\mathbf{x}) \ln f_{\mathrm{X}}(\mathbf{x})\, d\mathbf{x} \geq \int f_{\mathrm{X}}(\mathbf{x}) \ln g_{\mathrm{X}}(\mathbf{x})\, d\mathbf{x} \tag{F.8}$$

which is the version of the Jensen's inequality that we will use here.

## F.2. The EM Algorithm

The derivation of the EM algorithm presented here is a modification of the one used by Feder and Weinstein [20]. Considerably more detail and proofs are in the paper by Laird *et al.* [44].

Let's define **Z** as the observed data, **X** as the unobserved data and θ a parameter vector. We can express the densities as

$$f_{\mathrm{XZ}}(\mathbf{x}, \mathbf{z}; \theta) = f_{\mathrm{X/Z=z}}(\mathbf{x}; \theta)\, f_{\mathrm{Z}}(\mathbf{z}; \theta) \tag{F.9}$$

where $f_{\mathrm{XZ}}(\mathbf{x}, \mathbf{z}; \theta)$ is the *joint pdf* of the complete data, $f_{\mathrm{X/Z=z}}(\mathbf{x}; \theta)$ is the conditional pdf of the unobserved data **X** given the observed data **z** and $f_{\mathrm{Z}}(\mathbf{z}; \theta)$ is the pdf of the observed data **z**.

Now taking logarithms of (F.9) we can obtain:

$$\ln f_{\mathbf{Z}}(\mathbf{z};\theta) = \ln f_{\mathbf{XZ}}(\mathbf{x},\mathbf{z};\theta) - \ln f_{\mathbf{X/Z=z}}(\mathbf{x};\theta) \tag{F.10}$$

Let's take conditional expectations given $\mathbf{Z} = \mathbf{z}$ and the parameter $\theta'$ on (F.10):

$$\ln f_{\mathbf{Z}}(\mathbf{z};\theta) = E\{\ln f_{\mathbf{XZ}}(\mathbf{x},\mathbf{z};\theta) \,|\, \mathbf{Z} = \mathbf{z};\theta'\} - E\{\ln f_{\mathbf{X/Z=z}}(\mathbf{x};\theta) \,|\, \mathbf{Z} = \mathbf{z};\theta'\} \tag{F.11}$$

and define for convenience the following quantities:

$$L(\theta) = \ln f_{\mathbf{Z}}(\mathbf{z};\theta) \tag{F.12}$$

$$U(\theta,\theta') = E\{\ln f_{\mathbf{XZ}}(\mathbf{x},\mathbf{z};\theta) \,|\, \mathbf{Z} = \mathbf{z};\theta'\} \tag{F.13}$$

$$V(\theta,\theta') = E\{\ln f_{\mathbf{X/Z=z}}(\mathbf{x};\theta) \,|\, \mathbf{Z} = \mathbf{z};\theta'\} \tag{F.14}$$

so that (F.11) can be written in a compact form as:

$$L(\theta) = U(\theta,\theta') - V(\theta,\theta') \tag{F.15}$$

Applying the Jensen's inequality of (F.8) to (F.14) it is clear that

$$V(\theta,\theta') \le V(\theta',\theta') \tag{F.16}$$

Hence if we find another value of the parameter vector $\theta$ that makes

$$U(\theta,\theta') > U(\theta',\theta') \tag{F.17}$$

then a combination of (F.15), (F.16) and (F.17) lead to

$$L(\theta) > L(\theta') \tag{F.18}$$

With these relationships we have converted the problem of maximizing the log-likelihood $L(\theta)$ to the one of maximizing the function $U(\theta,\theta')$ defined by (F.13). The EM algorithm starts with an initial estimate for $\hat{\theta}^{(0)}$, and denote $\hat{\theta}^{(n)}$ the current estimate of $\theta$ after $n$ iterations. The iteration can be described in two steps:

1. *Estimate* $U(\theta,\hat{\theta}^{(n)})$
2. *Maximize* $U(\theta,\hat{\theta}^{(n)})$ to obtain $\hat{\theta}^{(n+1)}$

It is proved in (Laird *et al.* [44] that if $U(\theta,\theta')$ is continuous in both $\theta$ and $\theta'$, the algorithm converges to stationary point of the log-likelihood function where the maximization ensures that each iteration increases the likelihood. As in any "hill-climbing" method, the algorithm may converge to a local maximum rather than to the global maximum.

# Appendix G
# ML Estimation of Noise and Spectral Tilt

In this appendix we apply the EM algorithm described in Appendix F to the problem of maximum likelihood estimation of the noise $\mathbf{n}$ and spectral equalization $\mathbf{q}$ given the set of observations $\{\mathbf{z}_i;\ \ i = 0, 1, \ldots, N{-}1\}$. The unobserved data will be the mixture $k$ and the correction vectors $\{\mathbf{r}(\mathbf{x}_i)\}$. With these definitions, the function $U(\mathbf{n}, \mathbf{q}, \mathbf{n}', \mathbf{q}')$ defined in (F.13) has the form:

$$U = \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} \int p(\mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k \,|\, \mathbf{z}_i, \mathbf{n}', \mathbf{q}') \ln p(\mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k, \mathbf{z}_i \,|\, \mathbf{n}, \mathbf{q})\, d\mathbf{x}_i\, d\mathbf{r}(\mathbf{x}_i) \tag{G.1}$$

where the first term in the integral has the form

$$p(\mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k \,|\, \mathbf{z}_i, \mathbf{n}', \mathbf{q}') = \frac{p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k \,|\, \mathbf{n}', \mathbf{q}')}{p(\mathbf{z}_i \,|\, \mathbf{n}', \mathbf{q}')} \tag{G.2}$$

whose numerator can be expressed as

$$p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k \,|\, \mathbf{n}', \mathbf{q}') = p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k, \mathbf{n}', \mathbf{q}')\, p(\mathbf{r}(\mathbf{x}_i) \,|\, \mathbf{x}_i, k, \mathbf{n}', \mathbf{q}')\, p(\mathbf{x}_i, k \,|\, \mathbf{n}', \mathbf{q}')$$

$$= p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}(\mathbf{x}_i), k, \mathbf{n}', \mathbf{q}')\, \delta(\mathbf{r}(\mathbf{x}_i) - \mathbf{r}'(\mathbf{x}_i))\, p(\mathbf{x}_i \,|\, k)\, P_i[k] \tag{G.3}$$

by using the facts that the correction vector given $\mathbf{n}'$, $\mathbf{q}'$, $\mathbf{x}_i$ and mixture $k$ is $\mathbf{r}'(\mathbf{x}_i)$. Integration on $\mathbf{x}_i$ and $\mathbf{r}(\mathbf{x}_i)$, and summation on $k$ in Equation (G.3) yields the denominator in (G.2):

$$p(\mathbf{z}_i \,|\, \mathbf{n}', \mathbf{q}') = \sum_{k=0}^{K-1} P_i[k] \int p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k, \mathbf{n}', \mathbf{q}')\, p(\mathbf{x}_i \,|\, k)\, d\mathbf{x}_i \tag{G.4}$$

So that (G.1) takes on the new form

$$U = \sum_{i=0}^{N-1} \frac{\displaystyle\sum_{k=0}^{K-1} P_i[k] \int p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k \,|\, \mathbf{n}', \mathbf{q}') p(\mathbf{x} \,|\, k) \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k \,|\, \mathbf{n}, \mathbf{q})\, d\mathbf{x}_i}{\displaystyle\sum_{k=0}^{K-1} P_i[k] \int p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k \,|\, \mathbf{n}', \mathbf{q}') p(\mathbf{x}_i \,|\, k)\, d\mathbf{x}_i} \tag{G.5}$$

Also, the term with the logarithm in (G.5) can be expressed as

$$p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k \,|\, \mathbf{n}, \mathbf{q}) = p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k, \mathbf{n}, \mathbf{q})\, p(\mathbf{x}_i \,|\, k)\, P_i[k] \tag{G.6}$$

where

$$\ln p(\mathbf{z}_i \,|\, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k, \mathbf{n}, \mathbf{q}) = \alpha' - \frac{1}{2}(\mathbf{z}_i - \mathbf{q} - \mathbf{x}_i - \mathbf{r}'(\mathbf{x}_i))^T C_k^{-1} (\mathbf{z}_i - \mathbf{q} - \mathbf{x}_i - \mathbf{r}'(\mathbf{x}_i)) \tag{G.7}$$

or alternatively

$$\ln p(\mathbf{z}_i | \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k, \mathbf{n}, \mathbf{q}) = \alpha' - \frac{1}{2}(\mathbf{z}_i - \mathbf{n} - \mathbf{s}'(\mathbf{x}_i))^T C_k^{-1} (\mathbf{z}_i - \mathbf{n} - \mathbf{s}'(\mathbf{x}_i)) \qquad (G.8)$$

where we have used the **s** corrections instead of the **r** as they are uniquely related to each other. Taking logarithms, partial derivatives with respect to **q** in (G.6) and using (G.7) and (G.8) it can be obtained:

$$\frac{\delta}{\delta \mathbf{q}} \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}, \mathbf{q}) = -C_k^{-1} (\mathbf{z}_i - \mathbf{q} - \mathbf{x}_i - \mathbf{r}'(\mathbf{x}_i)) \qquad (G.9)$$

$$\frac{\delta}{\delta \mathbf{n}} \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}, \mathbf{q}) = -C_k^{-1} (\mathbf{z}_i - \mathbf{n} - \mathbf{s}'(\mathbf{x}_i)) \qquad (G.10)$$

Now assuming that the component due to the mixture 0 does not depend on **q**, which is reasonable if mixture 0 corresponds to the noise event:

$$\frac{\delta}{\delta \mathbf{q}} \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), 0 | \mathbf{n}, \mathbf{q}) = 0 \qquad (G.11)$$

and combining (G.5) and (G.9) we obtain

$$\frac{\delta U}{\delta \mathbf{q}} = \sum_{i=0}^{N-1} \frac{\sum_{k=1}^{K-1} P_i[k] \int p(\mathbf{z}_i | \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}', \mathbf{q}') p(\mathbf{x}_i | k)\, C_k^{-1} (\mathbf{z}_i - \mathbf{q} - \mathbf{x}_i - \mathbf{r}'(\mathbf{x}_i))\, d\mathbf{x}_i}{\sum_{k=1}^{K-1} P_i[k] \int p(\mathbf{z}_i | \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}', \mathbf{q}') p(\mathbf{x}_i | k)\, d\mathbf{x}_i} \qquad (G.12)$$

Making the derivative in (G.12) equal to zero and using again the approximation that the correction vectors are constant within a cluster:

$$\mathbf{r}'(\mathbf{x}_i) = \mathbf{r}'[k] \quad \forall i \qquad (G.13)$$

the integrals can be approximated as done in Appendix E:

$$0 = \sum_{i=0}^{N-1} \sum_{k=1}^{K-1} f_i[k] (\mathbf{q} + \mathbf{c}[k] + \mathbf{r}'[k] - \mathbf{z}_i) \qquad (G.14)$$

where $f_i[k]$ is given by:

$$f_i[k] = \frac{\dfrac{P_i[k]}{|C_k|^{1/2}} \exp(-d_i[k]/2)}{\sum_{l=0}^{K-1} \dfrac{P_i[l]}{|C_l|^{1/2}} \exp(-d_i[l]/2)} \quad \begin{cases} k = 0, 1, \ldots, K-1 \\ i = 0, 1, \ldots, N-1 \end{cases} \qquad (G.15)$$

Similarly, assuming that only the component due to the mixture 0 depends on **n**:

$$\frac{\delta}{\delta \mathbf{n}} \ln p(\mathbf{z}_i, \mathbf{x}_i, \mathbf{r}'(\mathbf{x}_i), k | \mathbf{n}, \mathbf{q}) = 0 \quad \forall\, k > 0 \qquad (G.16)$$

an analogous expression can be derived for **n**:

$$0 = \sum_{i=0}^{N-1} f_i[0] (\mathbf{n} + \mathbf{s}'[0] - \mathbf{z}_i) \qquad (G.17)$$

Finally by setting $\mathbf{s}'[0]$ to $\mathbf{0}$ according to (4.11) for the mixture 0 representing the noise event and solving (G.14) and (G.17), the ML estimates of **n** and **q** can be obtained as follows:

$$\hat{\mathbf{n}} = \frac{\sum_{i=0}^{N-1} f_i[0]\, \mathbf{z}_i}{\sum_{i=0}^{N-1} f_i[0]} \tag{G.18}$$

$$\hat{\mathbf{q}} = \frac{\sum_{i=0}^{N-1} \sum_{k=1}^{K-1} f_i[k]\, [\mathbf{z}_i - \mathbf{c}[k] - \mathbf{r}'[k]]}{\sum_{i=0}^{N-1} \sum_{k=1}^{K-1} f_i[k]} \tag{G.19}$$

Equations (G.18) and (G.19) give improved estimates of **n** and **q**, in the sense of a higher likelihood.

# Appendix H
# The Vocabulary and the Pronunciation Dictionary

| | |
|---|---|
| A | EY |
| AND | EH N DD |
| APOSTROPHE | AX P AA S T R OW F IY |
| APRIL | EY P R L |
| AREA | EH R IY AX |
| AUGUST | AO G AX S TD |
| B | B IY |
| C | S IY |
| CODE | K OW DD |
| D | D IY |
| DECEMBER | D IX S EH M B ER |
| E | IY |
| EIGHT | EY TD |
| EIGHTEEN | EY T IY N |
| EIGHTEENTH | EY T IY N TH |
| EIGHTH | EY TH |
| EIGHTY | EY DX IY |
| ELEVEN | AX L EH V IH N |
| ELEVENTH | AX L EH V IH N TH |
| ENTER | EH N T ER |
| ERASE | IX R EY S |
| F | EH F |
| FEBRUARY | F EH B Y UW EH R IY |
| FIFTEEN | F IH F T IY N |
| FIFTEENTH | F IH F T IY N TH |
| FIFTH | F IH F TH |
| FIFTY | F IH F T IY |
| FIRST | F ER S TD |
| FIVE | F AY V |
| FORTY | F AO R DX IY |
| FOUR | F AO R |
| FOURTEEN | F AO R T IY N |
| FOURTH | F AO R TH |
| G | JH IY |
| GO | G OW |
| H | EY CH |
| HALF | HH AE F |
| HALL | HH AA L |
| HELP | HH EH L PD |
| HUNDRED | HH AH N D R AX DD |
| I | AY |
| J | JH EY |
| JANUARY | JH AE N Y UW EH R IY |

| | |
|---|---|
| **JULY** | **JH AX L AY** |
| **JUNE** | **JH UW N** |
| **K** | **K EY** |
| **L** | **EH L** |
| **LANE** | **L EY N** |
| **M** | **EH M** |
| **MARCH** | **M AA R CH** |
| **MAY** | **M EY** |
| **MEMORY** | **M EH M R IY** |
| **N** | **EH N** |
| **NINE** | **N AY N** |
| **NINETEEN** | **N AY N T IY N** |
| **NINETY** | **N AY N DX IY** |
| **NINTH** | **N AY N TH** |
| **NO** | **N OW** |
| **O** | **OW** |
| **OCTOBER** | **AA KD T OW B ER** |
| **OF** | **AX V** |
| **OH** | **OW** |
| **ONE** | **W AH N** |
| **P** | **P IY** |
| **Q** | **K Y UW** |
| **R** | **AA R** |
| **REPEAT** | **R IX P IY TD** |
| **RUBOUT** | **R AH B AW TD** |
| **S** | **EH S** |
| **SECOND** | **S EH K AX N DD** |
| **SEPTEMBER** | **S EH PD T EH M B ER** |
| **SEVEN** | **S EH V AX N** |
| **SEVENTEEN** | **S EH V AX N T IY N** |
| **SEVENTH** | **S EH V AX N TH** |
| **SEVENTY** | **S EH V AX N DX IY** |
| **SIL** | **SIL** |
| **SIX** | **S IH K S** |
| **SIXTEEN** | **S IH K S T IY N** |
| **SIXTEENTH** | **S IH K S T IY N TH** |
| **SIXTH** | **S IH K S TH** |
| **SIXTY** | **S IH K S T IY** |
| **START** | **S T AA R TD** |
| **STOP** | **S T AA PD** |
| **T** | **T IY** |
| **TEN** | **T EH N** |
| **THIRD** | **TH ER DD** |
| **THIRTIETH** | **TH ER DX IY IX TH** |
| **THIRTY** | **TH ER DX IY** |
| **THOUSAND** | **TH AW Z AX N DD** |
| **THREE** | **TH R IY** |
| **TWELFTH** | **T W EH L F TH** |
| **TWELVE** | **T W EH L V** |
| **TWENTIETH** | **T W EH N IY IX TH** |
| **TWENTY** | **T W EH N IY** |
| **TWO** | **T UW** |
| **U** | **Y UW** |
| **V** | **V IY** |
| **W** | **D AH B AH L Y UW** |
| **WEAN** | **W IY N** |
| **X** | **EH K S** |

| | |
|---|---|
| **Y** | **W AY** |
| **YES** | **Y EH S** |
| **Z** | **Z IY** |
| **ZERO** | **Z IY R OW** |

# References


[1] A. Acero and R. M. Stern.
Environmental Robustness in Automatic Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 849-852. April, 1990.

[2] B. S. Atal.
Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification.
*Journal of the Acoustical Society of America* 55:1304-1312, Jun., 1974.

[3] M. Berouti, R. Schwartz and J. Makhoul.
Enhancement of Speech Corrupted by Acoustic Noise.
In J. S. Lim (editor), *Signal Processing*. Volume 1: *Speech Enhancement*, pages 69-73. Prentice Hall, Englewood Cliffs, NJ, 1983.

[4] B. Bogert, M. Healy and J. Tukey.
The Quefrency Alanysis of Time Series for Echoes.
In *Proc. Symp. on Time Series Analysis*, pages 209-243. J. Wiley, 1963.

[5] S. F. Boll.
Suppression of Acoustic Noise in Speech Using Spectral Subtraction.
*IEEE Trans. Acoustics, Speech and Signal Processing* 27(2):113-120, April, 1979.

[6] S. Boll, J. Porter and L. G. Bahler.
Robust Syntax Free Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 179-182. 1988.

[7] Y. Chen.
Cepstral Domain Talker Stress Compensation for Robust Speech Recognition.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-36:433-439, Apr., 1988.

[8] Chow, Y.L., Dunham, M.O., Kimball, O.A., Krasner, M.A., Kubala, G.F., Makhoul, J., Roucos, S., Schwartz, R.M.
BYBLOS: The BBN Continuous Speech Recognition System.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX*, pages 89-92. April, 1987.

[9] J. R. Cohen.
Application of an Auditory Model to Speech Recognition.
*Journal of the Acoustical Society of America* 85:2623-2629, Jun., 1989.

[10] D. Van Compernolle.
Spectral Estimation Using a Log-Distance Error Criterion Applied to Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 258-261. May, 1989.

[11] D. Van Compernolle.
Noise Adaptation in a Hidden Markov Model Speech Recognition System.
*Computer Speech and Language* , 1989.

[12] D. Van Compernolle.
Switching Adaptive Filters for Enhancing Noisy and Reverberant Speech from Microphone Array Recordings.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 833-836. April, 1990.

[13] S. B. Davis and P. Mermelstein.
Comparison of Parametric representations for Monosyllabic Word Recognition in Continuously Spoken Sentences.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-28(4):375-366, Aug., 1980.

[14] Y. Ephraim and D. Malah.
Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-32(6):1109-1121, Dec., 1984.

[15] Y. Ephraim and D. Malah.
Speech Enhancement Using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-33(2):443-445, Apr., 1985.

[16] Y. Ephraim, D. Malah and B. H. Huang.
Speech Enhancement Based upon Hidden Markov Modeling.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 353-356. May, 1989.

[17] Y. Ephraim.
A Minimum Mean Square Error Approach for Speech Enhancement.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 829-832. April, 1990.

[18] A. Erell and M. Weintraub.
Spectral Estimation for Noise Robust Speech Recognition.
In *Proc. Speech and Natural Language Workshop, Cape Cod, MA*. Morgan Kaufmann, Oct., 1989.

[19] A. Erell and M. Weintraub.
Estimation Using Log-Spectral-Distance Criterion for Noise-Robust Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 853-856. April, 1990.

[20] M. Feder and E. Weinstein.
Parameter Estimation of Superimposed Signals Using the EM Algorithm.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-36:477-489, Apr., 1988.

[21] M. Feder, A. V. Oppenheim and E. Weinstein.
Maximum Likelihood Noise Cancellation Using the EM Algorithm.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-37(2):204-216, Feb., 1989.

[22] J. L. Flanagan, J. D. Johnston, R. Zahn and G.W. Elko.
Computer-steered Microphone Arrays for Sound Transduction in Large Rooms.
*Journal of the Acoustical Society of America* 78:1508-1518, Nov., 1985.

[23] S. Furui.
Cepstral Analysis Technique for Automatic Speaker Verification.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-29(2):254-272, Apr., 1981.

[24] S. Furui.
Unsupervised Speaker Adaptation Method Based on Hierarchical Spectral Clustering.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 286-289. May, 1989.

[25] O. Ghitza.
Auditory Nerve Representation Criteria for Speech Analysis/Synthesis.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-35:736-740, Jun., 1987.

[26] O. Ghitza.
Auditory Neural Feedback as a Basis for Speech Processing.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 91-94. 1988.

[27] L. Gillick.
Some Statistical Issues in the Comparison of Speech Recognition Algorithms.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 532-535. May, 1989.

[28] H. Gish, Y. Chow and J. R. Rohlicek.
Probabilistic Vector Mapping of Noisy Speech Parameters for HMM Word Spotting.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 117-120. April, 1990.

[29] V. Goncharoff and S. Chandran.
Adaptive Speech Modification by Spectral Warping.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 343-346. 1988.

[30] R. Gray, A. Buzo, A. Gray and Y. Matusyama.
Distance Measures for Speech Processing.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-24:380-391, Oct., 1976.

[31] Gray, R.M.
Vector Quantization.
*IEEE ASSP Magazine* 1(2):4-29, April, 1984.

[32] J. H. Hansen and M. A. Clements.
Constrained Iterative Speech Enhancement with Application to Automatic Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 561-564. 1988.

[33] J. H. Hansen.
Stress Compensation and Noise Reduction Algorithms for Robust Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 266-269. May, 1989.

[34] H. Hermansky.
Optimization of Perceptually-Based ASR Front-End.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 219-222. 1988.

[35] H. W. Hon and K. F. Lee.
On Vocabulary-Independent Speech Modeling.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 725-728. April, 1990.

[36] X. D. Huang, K. F. Lee and H. W. Hon.
On Semi-Continuous Hidden Markov Modeling.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 689-692. April, 1990.

[37] M. J. Hunt and C. Lefebvre.
Speaker Dependent and Independent Speech Recognition Experiments with an Auditory Model.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 215-218. 1988.

[38] M. J. Hunt and C. Lefebvre.
A comparison of Several Acoustic Representations for Speech Recognition with Degraded and Undegraded Speech.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 262-265. May, 1989.

[39] F. Itakura and T. Umezaki.
Distance Measure for Speech Recognition Based on the Smoothed Group Delay Spectrum.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX,* pages 1257-1260. 1987.

[40] N. S. Jayant and P. Noll.
*Digital Coding of Waveforms.*
Prentice Hall, 1984.

[41] B. H. Juang, L. R. Rabiner and J. G. Wilpon.
On the Use of Bandpass Liftering in Speech Recognition.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-35:947-954, Jul., 1987.

[42] J. C. Junqua and H. Wakita.
A Comparative Study of Cepstral Lifters and Distance Measures for All-Pole Models of Speech in Noise.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK,* pages 476-479. May, 1989.

[43] S. M. Kay.
*Modern Spectral Estimation.*
Prentice Hall, 1988.

[44] N.M. Laird, A.P. Dempster and D.B. Rubin.
Maximum Likelihood from Incomplete Data via the EM algorithm.
*Ann. Roy. Stat. Soc.* :1-38, Dec, 1987.

[45] K. F. Lee and H. W. Hon.
Large-Vocabulary Speaker-Independent Continuous Speech Recognition Using HMM.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY,* pages 123-126. 1988.

[46] K. F. Lee, H.W Hon, M.Y Hwang, S. Mahajan and R. Reddy.
The SPHINX Speech Recognition System.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK,* pages 445-448. May, 1989.

[47] K. P. Li and J. Porter.
Normalizations and Selection of Speech Segments for Speaker Recognition Scoring.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY,* pages 595-598. 1988.

[48] J. S. Lim and A. V. Oppenheim.
All-Pole Modeling of Degraded Speech.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-26:197-210, Jun., 1978.

[49] J. S. Lim.
*Speech Enhancement.*
Prentice-Hall, Englewood Cliffs, NJ, 1983.

[50] Linde, Y., Buzo, A., Gray, R.M.
An Algorithm for Vector Quantizer Design.
*IEEE Transactions on Communication* COM-28(1):84-95, January, 1980.

[51] R. P. Lippmann, E. A. Martin and D.B. Paul.
Multi-Style Training for Robust Isolated-Word Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX*, pages 705-708. April, 1987.

[52] D. Mansour and B. H. Juang.
A Family of Distortion Measures Based Upon Projection Operation for Robust Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 36-39. 1988.

[53] K. S. Min, D. Chien, S. Li and C. Jones.
Automated Two-Speaker Separation System.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 537-540. 1988.

[54] S. Morii.
Spectral Subtraction in the Sphinx System.
unpublished.
1988

[55] H. Murveit and M. Weintraub.
1000-Word Speaker-Independent Continuous-Speech Recognition Using Hidden Markov Models.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 115-118. 1988.

[56] A. J. Nadas, D. Nahamoo and M. A. Picheny.
Speech Recognition Using Noise-Adaptive Prototypes.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 517-520. 1988.

[57] A. J. Nadas, D. Nahamoo and M. A. Picheny.
Adaptive labeling: Normalization of Speech by Adaptive Transformations Based on Vector Quantization.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 521-524. 1988.

[58] J. A. Naylor and S. F. Boll.
Techniques for Suppression of an Interfering Talker in Co-Channel Speech.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX*, pages 205-208. 1987.

[59] Nocerino, F. K. Soong, L. R. Rabiner and D. H. Klatt.
Comparative Study of Several Distortion Measures for Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Atlanta, GA*, pages 25-28. Apr., 1985.

[60] H. Noda.
Frequency-Warped Spectral Distance Measures for Speaker Verification in Noise.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 576-579. 1988.

[61] A. V. Oppenheim and D. H. Johnson.
Discrete Representation of Signals.
*Proc. of the IEEE* (33):681-691, 1972.

[62] D. S. Pallett.
Benchmark Tests for DARPA Resource Management Performance Evaluations.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Glasgow, UK*, pages 536-539. May, 1989.

[63] D. Pallett, J. G. Fiscus and J. S. Garofolo.
DARPA Resource Management Benchmark Test Results, June 1990.
In *Proc. Speech and Natural Language Workshop, Hidden Valley, PA.* Morgan Kaufmann, Jun., 1990.

[64] A. Papoulis.
*Probability, Random Variables and Stochastic Processes.*
Mc Graw Hill, 1984.

[65] D. B. Paul.
A Speaker-Stress Resistant Isolated Word Recognizer.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX*, pages 713-716. 1987.

[66] J. Picone.
Continuous Speech Recognition Using Hidden Markov Models.
*IEEE ASSP Magazine* 7(3):26-41, July, 1990.

[67] J. E. Porter and S. F. Boll.
Optimal Estimators for Spectral Restoration of Noisy Speech.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, San Diego, CA*, pages 18A.2.1. May, 1984.

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

[69] Rabiner, L.R., Juang, B.H.
An Introduction to Hidden Markov Models.
*IEEE ASSP Magazine* 3(1):4-16, January, 1986.

[70] S. Seneff.
A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing.
*Journal of Phonetics* 16:55-76, Jan., 1988.

[71] Shikano, K.
*Evaluation of LPC Spectral Matching Measures for Phonetic Unit Recognition.*
Technical Report, Computer Science Department, Carnegie Mellon University, May, 1986.

[72] H. Silverman.
An Algorithm for Determining Talker Location using a Linear Microphone Array and Optimal Hyperbolic Fit.
In *Proc. Speech and Natural Language Workshop, Hidden Valley, PA.* Morgan Kaufmann, Jun., 1990.

[73] F. K. Soong and M. M. Sondhi.
A Frequency-Weighted Itakura Spectral Distortion Measure and its Application to Speech Recognition in Noise.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Dallas, TX*, pages 625-628. 1987.

[74] R. Stern and A. Acero.
Acoustical Pre-processing for Robust Speech Recognition.
In *Proc. Speech and Natural Language Workshop, Cape Cod, MA*, pages 311-318. Morgan Kaufmann, Oct., 1989.

[75] T. G. Stockham, T. M. Cannon and R. B. Ingebretsen.
Blind Deconvolution Through Digital Signal Processing.
*Proc. of the IEEE* 63(4):678-692, Apr., 1975.

[76] S. Tamura and A. Waibel.
Noise Reduction Using Connectionist Models.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 553-556. 1988.

[77] S. Tamura and M. Nakamura.
Improvements to the Noise Reduction Neural Network.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 825-828. April, 1990.

[78] Y. Tokhura.
A Weighted Cepstral Distance Measure for Speech Recognition.
*IEEE Trans. Acoustics, Speech and Signal Processing* ASSP-35:1414-1422, Oct., 1987.

[79] A. Varga, R. Moore, J. Bridle, K. Ponting and M. Russell.
Noise Compensation Algorithms for Use with Hidden Markov Model Based Speech Recognition.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 481-484. 1988.

[80] A. P. Varga and R. K. Moore.
Hidden Markov Model Decomposition of Speech and Noise.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 845-848. April, 1990.

[81] W. Ward.
Modeling Non-Verbal Sounds for Speech Recognition.
In *Proc. Speech and Natural Language Workshop, Cape Cod, MA*, pages 311-318. Morgan Kaufmann, Oct., 1989.

[82] R. Zelinski.
A Microphone Array with Adaptive Post-Filtering for Noise Reduction in Reverberant Rooms.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, New York, NY*, pages 2578-2581. 1988.

[83] V. Zue, J. Glass, D. Goodine, M. Phillips and S. Seneff.
The Summit Speech Recognition System: Phonological Modelling and Lexical Access.
In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Albuquerque, NM*, pages 49-52. April, 1990.

[84] E. Zwicker.
Subdivision of the Audible Frequency Range into Critical Bands.
*Journal of the Acoustical Society of America* (33):248-249, 1961.

# Table of Contents


Abstract 1
Acknowledgments 2
1. Introduction 3
1.1. Acoustical Environmental Variability and its Consequences 4
1.1.1. Input Level 4
1.1.2. Additive Noise 4
1.1.3. Spectral Tilt 5
1.1.4. Physiological Differences 5
1.1.5. Interference from Other Speakers 6
1.2. Previous Research in Signal Processing for Robust Speech Recognition 6
1.2.1. Techniques Based on an Autoregressive Model 6
1.2.2. Techniques Based on Manipulation of Distortion Measures 7
1.2.3. The Use of Auditory Models 8
1.2.4. Techniques Based on Short-Time Spectral Amplitude Estimation 9
1.2.5. Techniques Based on Mixture Densities 10
1.2.6. Other Techniques 11
1.2.7. Discussion 12
1.3. Towards Environment-Independent Recognition 12
1.3.1. Joint Compensation for Noise and Equalization 13
1.3.2. Processing in the Cepstral Domain: A Unified View 13
1.3.3. Measuring Performance Evaluation 14
1.4. Dissertation Outline 15
2. Experimental Procedure 16
2.1. An Overview of SPHINX 16
2.1.1. Signal Processing 16
2.1.2. Vector Quantization 18
2.1.3. Hidden Markov Models 18
2.1.4. Speech Units 19
2.2. The Census Database 19
2.2.1. Speaker Population 20
2.2.2. Database Contents 20
2.2.3. Alphanumeric Database 20
2.2.4. The Environment 21
2.2.5. The Recognition System 21

2.3. Objective Measurements 21
2.3.1. Measurements for Stationary Signals: SNR 22
2.3.2. Measurements for Nonstationary Signals: SEGSNR and MAXSNR 22
2.3.3. Frequency-Weighted SNR 23
2.3.4. A Proposed Solution: Average Speech and Noise Spectra 24
2.3.5. Discussion of SNR Measures 25
2.4. Baseline Recognition Accuracy 26
2.4.1. Error Analysis 27
2.5. Other Databases 28
2.5.1. Sennheiser HMD224 - Crown PCC160 29
2.5.2. Sennheiser HMD224 - Crown PZM6fs 30
2.5.3. Sennheiser HMD224 - Sennheiser 518 31
2.5.4. Sennheiser HMD224 - Sennheiser ME80 32
2.5.5. HME FM - Crown PCC160 33
2.6. Summary 34
3. Processing in the Frequency Domain 35
3.1. Multi-Style Training 35
3.2. Channel Equalization 37
3.3. Noise Suppression by Spectral Subtraction 38
3.3.1. Spectral Subtraction for Speech Enhancement 38
3.3.2. Noise Subtraction in Speech Recognition 40
3.3.3. Spectral Subtraction in the Logarithm Domain 40
3.4. Experiments with Sphinx 43
3.4.1. EQUAL Algorithm 44
3.4.2. PSUB Algorithm 46
3.4.3. MSUB Algorithm 48
3.4.4. MMSE1 Algorithm 48
3.4.5. Cascade of EQUAL and MSUB 54
3.4.6. Results and Discussion 57
3.5. Summary 59
4. Joint Compensation for Noise and Filtering 60
4.1. A Model of the Environment 60
4.2. Processing in the Frequency Domain: The MMSEN Algorithm 62
4.3. Processing in the Cepstral Domain: The SDCN Algorithm 65
4.4. Summary 71
5. CDCN Algorithm 72
5.1. Introduction to the CDCN Algorithm 74
5.1.1. Using Only Acoustic Information 75
5.1.2. Using Discrete Models 76
5.2. MMSE Estimator of the Cepstral Vector 76
5.3. ML Estimation of Noise and Spectral Tilt 78
5.4. Summary of the CDCN Algorithm 80
5.5. Implementation Details 81
5.6. Evaluation Results 81

5.7. Summary 85

6. Improving the Efficiency 86

6.1. Interpolated SDCN 86

6.2. Fixed CDCN 88

6.2.1. Estimating the Correction Vectors 90

6.3. Estimating the Environmental Parameters from Previous Utterances 94

6.4. Summary 95

7. Frequency Normalization 96

7.1. The Use of Mel-scale Parameters 96

7.2. Improving the Frequency Resolution 97

7.3. Variable Frequency Warping 100

7.4. Summary 103

8. Summary of Results 104

9. Conclusions 109

9.1. Contributions 109

9.2. Suggestions for Future Work 111

Appendix A. Glossary 114

A.1. Time Domain 114

A.2. Frequency Domain 114

A.3. Cepstral Domain 114

A.4. Indices 115

A.5. Probabilistic Models 115

Appendix B. Signal Processing in SPHINX 116

Appendix C. The Bilinear Transform 118

C.1. Cascade of Warping Stages 119

Appendix D. Spectral Estimation Issues 120

Appendix E. MMSE Estimation in the CDCN Algorithm 121

E.1. The Conditional Probability in the CDCN Algorithm 121

E.2. Gaussian Decomposition 122

Appendix F. Maximum Likelihood via the EM Algorithm 124

F.1. Jensen’s Inequality 124

F.2. The EM Algorithm 125

Appendix G. ML Estimation of Noise and Spectral Tilt 127

Appendix H. The Vocabulary and the Pronunciation Dictionary 130

References 133

# List of Figures


| | | |
|---|---|---|
| **Figure 2-1:** | **Block diagram of SPHINX front-end.** | **17** |
| **Figure 2-2:** | **Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PZM6fs microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency. It can also be seen that the Crown PZM6sf produces greater spectral tilt.** | **25** |
| **Figure 2-3:** | **Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PCC160 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.** | **29** |
| **Figure 2-4:** | **Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Crown PZM6fs microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.** | **30** |
| **Figure 2-5:** | **Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Sennheiser 518 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.** | **31** |
| **Figure 2-6:** | **Average speech and noise spectra from the Alphanumeric database obtained using the headset-mounted Sennheiser HMD224 Microphone and the Sennheiser ME80 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.** | **32** |

| | | |
|---|---|---|
| **Figure 2-7:** | **Average speech and noise spectra from the Alphanumeric database obtained using the HME FM Microphone and the Crown PCC160 microphone. The separation of the two curves in each panel provides an indication of signal-to-noise ratio for each microphone as a function of frequency.** | **33** |
| **Figure 3-1:** | **Spectral subtraction curve for** $\hat{\mathbf{N}}(\omega_k) = 25\,\mathrm{dB}$ **and** $\mathbf{X}_{th}(\omega_k) = 10\,\mathrm{dB}$**.** | **42** |
| **Figure 3-2:** | **Model of the degradation.** | **43** |
| **Figure 3-3:** | **3D Spectrogram of the utterance *yes* recorded with the CLSTK microphone with no processing.** | **44** |
| **Figure 3-4:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with no processing. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **45** |
| **Figure 3-5:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with EQUAL algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **47** |
| **Figure 3-6:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with PSUB algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **49** |
| **Figure 3-7:** | **Amount of over and under subtraction used in the MSUB algorithm as a function of the instantaneous SNR.** | **50** |
| **Figure 3-8:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with MSUB algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **51** |
| **Figure 3-9:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with cascade of EQUAL and MMSE1 algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **53** |
| **Figure 3-10:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with the cascade of EQUAL and MMSE1 algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **55** |

| | | |
|---|---|---|
| **Figure 3-11:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with the cascade of EQUAL and MSUB algorithms. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **56** |
| **Figure 3-12:** | **Input-Output transformation curves for PSUB, MSUB and MMSE1. The channel SNR is defined as the log-power of the signal in a frequency band minus the log-power of the noise in that band. The transformation for MSUB is not a single curve but a family of curves that depend on the total SNR for a given frame.** | **58** |
| **Figure 4-1:** | **Model of the degradation.** | **61** |
| **Figure 4-2:** | **Comparison between the transformation curve MMSE fixed for all frequencies and the corresponding transformations for different frequencies: 0, 1, 2, 3, 4, 5, 6 and 8 kHz. The curves give the input-output relation between the SNR at a frequency band. It can be seen that more noise subtraction is done at low frequencies than at high frequencies. Also, low frequencies are attenuated more at high SNR, to compensate for spectral tilt.** | **64** |
| **Figure 4-3:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with MMSEN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **66** |
| **Figure 4-4:** | **Correction vector w[*i*] in Np as a function of the instantaneous SNR in dB. Note the different scale used for w[0] and w[1], as they are the correction vectors varying the most. Correction vector w[12] is not shown.** | **68** |
| **Figure 4-5:** | **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with SDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** | **69** |
| **Figure 4-6:** | **Word accuracy of the SDCN algorithm as a function of the number of correction vectors used when trained on the CLSTK and tested on the CRPZM. None means that no correction is applied (baseline), w[0] means that only c[0] was compensated, w[1] that both c[0] and c[1] were compensated, etc.** | **70** |
| **Figure 5-1:** | **CDCN estimates the noise n and channel equalization q that best transform the universal codebook into the ensemble of input frames of the current utterance.** | **72** |

**Figure 5-2:** **3D Spectrogram of the utterance *yes* recorded with the CLSTK microphone processed with the CDCN algorithm.** **82**

**Figure 5-3:** **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with CDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** **83**

**Figure 6-1:** **Variance of the difference vector between the CLSTK and the restored CRPZM speech for different input SNR of the CRPZM.** **92**

**Figure 6-2:** **3D Spectrogram of the utterance *yes* recorded with the CRPZM microphone with FCDCN algorithm. Spectra at times 190 ms (silence region), 400 ms (vowel) and 650 ms (fricative) are plotted for both the CLSTK and the CRPZM.** **93**

**Figure 7-1:** **Frequency mapping of the Bilinear Transform for different values of $\alpha$. Note that $\alpha = 0$ is equivalent to no warping while $\alpha = 0.8$ is a very severe warping of the frequency axis.** **98**

**Figure 7-2:** **Spectral analysis of the vowel /ih/. The first graph shows the FFT spectrum and the LPC spectrum of order 14. The second graph shows the LPC spectrum and its cepstral approximation with 32 and 12 coefficients. The third graph shows the warped spectrum plotted from 12 cepstral coefficients after the bilinear transform with $\alpha = 0.6$ when both 32 and 12 coefficients are used before the transform. The use of 12 coefficients before the warping removes the formant structure.** **99**

**Figure 7-3:** **Histogram of values of $\alpha$ for male and female speakers.** **102**

**Figure 8-1:** **Independent compensation for noise and filtering in the spectral domain. Comparison of the baseline performance (BASE), cepstral equalization (EQUAL), Power Spectral Subtraction (PSUB), Minimum Mean Squared Error with one curve for all frequencies (MMSE1) and Magnitude Spectral Subtraction (MSUB) when trained on the CLSTK microphone and tested on the CRPZM. The broken line represents the word accuracy of the system trained and tested on the CRPZM.** **106**

**Figure 8-2:** **Comparison of algorithms that perform independent compensation for both noise and filtering (EQ+MMSE1, EQ+MSUB) versus an algorithm that performs joint compensation for noise and filtering (MMSEN) in the spectral domain. The Figure shows the word accuracy of SPHINX when trained on the CLSTK microphone and tested on the CRPZM. The** **106**

**broken line represents the word accuracy of the system trained and tested on the CRPZM.**

**Figure 8-3: Comparison of the performance of algorithms in the spectral domain (MMSEN) and the cepstral domain (SDCN,FCDCN). All these algorithms perform joint compensation for noise and filtering but they also require stereo data. MMSEN uses a minimum mean squared error criterion with one transformation curve per frequency component, and it is described in Chapter 4. SDCN, SNR-Dependent Cepstral Normalization , and the FCDCN, Fixed Codeword-Dependent Cepstral Normalization are described in Chapters 4 and 6 respectively. SPHINX was trained on the CLSTK microphone and tested on the CRPZM. The broken line represents the word accuracy of the system trained and tested on the CRPZM. 107**

**Figure 8-4: Comparison of algorithms that adapt to new acoustical environments (ISDCN and CDCN) algorithms when trained on the CLSTK microphone and tested on the CRPZM. BASE represents no processing, ISDCN is the Interpolated SNR-Dependent Cepstral Normalization described in Chapter 6 and CDCN is the Codeword-Dependent Cepstral Normalization described in Chapter 5. ISDCN and CDCN perform also joint compensation for noise and filtering. The broken line represents the word accuracy of the system trained and tested on the CRPZM. 107**

**Figure B-1: Pascal window for $\tau = 1500$ 116**

**Figure C-1: Bilinear transform algorithm as a linear filtering operation. By having as the input to this stage of filters a time-reversed cepstrum sequence, we can obtain the corresponding warped coefficients as the outputs of these filters at time $n = 0$. 118**

**Figure F-1: Functions $f(y) = y - 1$ and $f(y) = \ln(y)$. 124**

# List of Tables


| | | |
|---|---|---|
| **Table 2-1:** | **Analysis of different SNR measures for the census database. In all cases a 20 ms Hamming window was used. All the figures are computed as the average in dB across all the utterances in the database of different measures: maximum signal energy in the utterance for MAXSNR, the average signal energy for SNR, the average log-energy of the signal for SEGSNR. AVGSPT is the average separation of curves in Figure 2-2** | **26** |
| **Table 2-2:** | **Baseline recognition rate of the Sphinx system when trained and tested on the census vocabulary using each of the two microphones.** | **27** |
| **Table 2-3:** | **Analysis of causes of "new" errors introduced by use of the Crown PZM microphone.** | **27** |
| **Table 2-4:** | **Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Crown PCC160.** | **29** |
| **Table 2-5:** | **Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Crown PZM6fs.** | **30** |
| **Table 2-6:** | **Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Sennheiser 518.** | **31** |
| **Table 2-7:** | **Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the Sennheiser HMD224 and the Sennheiser ME80.** | **32** |
| **Table 2-8:** | **Comparison of MAXSNR, SNR and SEGSNR measurements for the alphanumeric database recorded with the HME FM and the Crown PCC160.** | **33** |
| **Table 3-1:** | **Comparison of the baseline performance of the system under different training conditions: close-talking microphone, Crown PZM microphone and multi-style training. Testing is done for the two microphones.** | **36** |
| **Table 3-2:** | **Baseline recognition accuracy for the CLSTK and CRPZM.** | **46** |
| **Table 3-3:** | **Comparison of the baseline performance and the EQUAL spectral equalization algorithm. EQUAL was only applied to the CRPZM.** | **48** |

| | | |
|---|---|---|
| **Table 3-4:** | **Comparison of the baseline performance with the PSUB power subtraction algorithm. PSUB was only applied to the CRPZM.** | **50** |
| **Table 3-5:** | **Comparison of the baseline performance with the PSUB power subtraction and the MSUB magnitude subtraction algorithms. PSUB and MSUB were only applied to the CRPZM.** | **52** |
| **Table 3-6:** | **Comparison of baseline performance with the PSUB, MSUB and MMSE1 algorithms. These algorithms were only applied to the CRPZM.** | **54** |
| **Table 3-7:** | **Performance of different equalization and spectral subtraction algorithms. All the algorithms were applied only to the CRPZM.** | **57** |
| **Table 4-1:** | **Performance of the MMSEN compared with the Baseline and the MMSE1 and MSUB algorithms.** | **65** |
| **Table 4-2:** | **Performance of the MMSEN and SDCN algorithms when compared with the baseline.** | **70** |
| **Table 5-1:** | **Comparison of recognition accuracy of SPHINX with no processing, SDCN and CDCN algorithms. The system was trained and tested using all combinations of the CLSTK and CRPZM microphones.** | **82** |
| **Table 5-2:** | **Analysis of performance of SPHINX for the baseline and the CDCN algorithm. Two microphones were recorded in stereo in each case. The microphones compared are the Sennheiser HMD224, 518, ME80, the Crown PZM6FS and PCC160, and the HME microphone. Training was done with the Sennheiser HMD224 in all cases.** | **84** |
| **Table 6-1:** | **Performance of the ISDCN algorithm as compared with the baseline and the CDCN. The algorithm was applied to both CLSTK and CRPZM and training was done with the processed CLSTK.** | **88** |
| **Table 6-2:** | **Analysis of performance of SPHINX for the baseline and the CDCN, ISDCN and FCDCN algorithms. Two microphones were recorded in stereo in each case. The microphones compared are the Sennheiser HMD224 (CLSTK), Crown PZM6FS, Crown PCC160, Sennheiser 518 and SennheiserME80. Training was done with the Sennheiser HMD224 (CLSTK) from the census database in all cases. The correction vectors for the FCDCN were estimated from each stereo database and are different for every experiment.** | **89** |
| **Table 6-3:** | **Analysis of performance of SPHINX for the baseline and the CDCN and ISDCN algorithms. The Crown PCC160 and the HME FM microphone were recorded in stereo using training with the Sennheiser HMD224 (CLSTK) from the census database.** | **90** |

| | | |
|---|---|---|
| **Table 6-4:** | **Performance of the Fixed CDCN algorithm as compared with the baseline and the CDCN. This algorithm is only applied to the CRPZM using the training models for the baseline case.** | **94** |
| **Table 6-5:** | **Performance of the ISDCN algorithm when the environmental parameters are computed from the same utterance (ISDCN) or the previous utterance (ISDCNprev). The performance is compared to the baseline (BASE) and the FCDCN algorithm.** | **95** |
| **Table 7-1:** | **Effect of the use of 12 and 32 cepstral coefficients before the bilinear transform in SPHINX. The number of coefficients after frequency warping was 12 in both cases.** | **100** |
| **Table 7-2:** | **Comparison of the performance of the variable frequency warping with the baseline.** | **101** |
| **Table 7-3:** | **Comparison of the performance of the variable frequency warping with the baseline.** | **102** |
| **Table 8-1:** | **Performance of different normalization algorithms.** | **104** |