Environmental Adaptation for Robust Speech Recognition

Fu-Hua Liu
June 28, 1994

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

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Electrical Engineering

Contents

List of Figures

List of Tables

Abstract

Lack of robustness with respect to environmental variability is a continuing problem for speech recognition. Many studies have shown that automatic speech recognition systems perform poorly when there are differences in the acoustics of the training and testing environment. Several approaches have previously been considered to compensate for environmental variability, including techniques based on autoregressive analysis, the use of auditory models, and the use of array processing, among many other approaches.

This dissertation describes a number of new algorithms that improve the ability of speech recognition systems to adapt to new acoustical environments. These new testing environments are assumed to differ from the training environment because of the presence of both unknown additive noise and distortion from unknown linear filtering. The algorithms are based on previous research in which significant environmental robustness had been achieved by modifying the cepstral coefficients that are input as features to speech recognition systems. The present work extends the previous results along the dimensions of improved recognition accuracy, reduced dependence on specialized training data, reduced computational cost, and greater integration of environmental compensation into the matching algorithm of the speech decoder.

Environmental compensation is generally accomplished by the application of one of an ensemble of additive corrections to either the features that are input to the recognition system, or to the internal representation of speech inside the recognition system itself. The exact compensation is time varying. For each 20-ms speech segment the choice of compensation vector depends either on physical attributes such as the instantaneous signal-to-noise ratio, or on the putative identity of the phoneme during that segment as hypothesized by the speech decoder. The actual values of the compensation vectors are determined by frame-by-frame comparisons of large numbers of cepstral vectors of speech that is simultaneously recorded in the training environment and in one of a number of prototype secondary environments. Compensation is performed by first estimating which of the prototype environments most closely resembles the testing environment, and then by applying the compensation vectors that are appropriate for that environment.

The new algorithms are evaluated in terms of their effectiveness in improving environmental robustness and their computational complexity, among other attributes. It is found that further increases in robustness can be obtained by combining algorithms that process features that are input to the system with algorithms that modify the system's internal representation of speech. Linear interpolation of compensation vectors from different environments is generally helpful when the system was tested in an environment that was not one of the prototypes used to develop compensation vectors. In a standard ARPA evaluation of a 5000-word system that recognized sentences recorded in unknown environments, combination of these techniques typically decreases the rate of word errors by 66% compared to no environmental processing at all, and by 40% when the standard technique of cepstral mean normalization is included in the baseline system.

Acknowledgments

First and foremost, I would like to thank Richard M. Stern, my thesis advisor, for his support, advice, and encouragement. His exceptional scientific approach and experience have been an inspiration to me in the years of my thesis research. He is not only an academic mentor who advises me on research, but also a great tutor with enthusiasm and patience to enlighten me on various aspects. Without his generous support and assistance, this thesis work would not have been what it is now.

I would also like to thank other members of my thesis committee, Raj Reddy, Alejandro Acero, and Vijaya Kumar. I am highly indebted to Raj for his generous financial support during my graduate years at CMU and for the best CMU speech group he presents. Alejandro has followed my thesis work closely for the past few years. I am truly grateful to him for many precious suggestions and advice to my research. Kumar also provides insightful comments and valuable feedback for this dissertation.

I would also like to thank Kai-Ku Lee who introduced me to the CMU speech group and provided me with a chance to work in such an unsurpassed research environment. I would also like to thank Xuedong Huang for many fruitful discussions. I am also indebted to Mei-Yuh Hwang for her enthusiasm to answer my questions about many mysteries about the SPHNX-II system.

Fellow graduate students of Richard Stern's, previous or current, Pedro Moreno, Yoshiaki Ohshima, Tom Sullivan, Nobutoshi Hanai, Uday Jain, Matthew Siegler, and Sammy Tao, have been an excellent source to seek feedback and suggestions to my ideas and experiments. I am particular indebted to Matthew Siegler for his patience to proofread my thesis when I wrote it. Without his help, it would have taken me a few more months to finish the writing.

I would also like to thank members of the CMU Speech Group. Hsiao-Wuen Hon and Fil Alleva had helped me with the recognition system in my early years of thesis work. Bob Weide never hesitates to respond to me regarding database and hardware resources. Eric Thayer and Ravishankar Mosur helped me understand more about the search algorithms in the final months. I am also grateful to other members of the speech group for making CMU a wonderful environment for speech research. I am very proud to have opportunities to work with all these bright minds during my years at CMU.

I am extremely indebted to my parents for their support and for all that they have taught me. Their love and encouragement have been a significantly supportive power to me over the years. I would also like to thank my parents-in-law for their support and, more importantly, for allowing me to take their lovely daughter, Pei-Chi, for my wife. Pei-Chi patiently supported me when I worked on this thesis. At last, I am glad to have this chance to devote this thesis to her for the love, care, and thoughtfulness she has given me throughout these long years.

Chapter 1
Introduction

While automatic speech recognition has been a goal of research for many years, it has a long history of being one of the most difficult challenges. It will still take many more years to design an intelligent machine that can understand spoken discourse on any subject by all speakers in all environments. Early research in speech recognition attained acceptable performances only by imposing constraints in the task domain such as speaker dependence, isolated words, small vocabulary, constrained grammar, or the use of a quiet recording environment.

During the past few years, progress in speech recognition technology has been made in addressing some of these constraints enabling the development of practical applications of speech recognition technology. Many recently-developed speech recognition systems [*e.g.* 40, 43,50, 25, 110] have addressed some of these challenges in specific domains. For example, speaker-independent continuous speech recognition for vocabulary sizes of 5,000 to 20,000 words using sophisticated language models have been demonstrated by a number of systems, including the CMU SPHINX-II system. This system has been applied to the standard ARPA domains of dictation from the World Street Journal and the Air Travel Information Service task.

As automatic speech recognition systems are finding their way into practical applications it is becoming increasingly clear that they must exhibit "robustness" to accommodate a variety of conditions in the field. Ideally an unconstrained system would be robust to the following challenges (among others):

- **Speaker Variability**: speaker dependence versus speaker independence.
- **Speaking Style Variability**: isolated-word recognition versus continuous speech recognition.
- **Vocabulary Size Variability**: small-vocabulary tasks versus large-vocabulary tasks.
- **Domain Variability**: vocabulary dependence versus vocabulary independence, and domain-dependent grammar versus domain-independent grammar.
- **Environment Variability**: environment-dependence versus environment-independence.

Each of these above issues constitutes a fundamental difficult challenge for researchers to solve before a speech recognition system can be deployed for any application in an unconstrained fashion. In this thesis we focus on the last issue, the environmental robustness of speech recognition, and we describe several algorithms to adapt to a variety of acoustical environments.

1.1. Approaches to Overcoming Environmental Variability

It has been observed that speech recognition accuracy degrades dramatically when the acoustical characteristics of the training and testing environments differ. In the past, many speech recognition systems were designed under the assumption of ideal acoustical ambience, ignoring the presence sources of variability such as noise and distortion. This lack of robustness with respect to environmental variability remains a continuing problem with current speech recognition technology.

For example, Acero [1] showed that a speaker-independent continuous speech recognition system with a baseline performance of 85% word accuracy could only achieve 19% word accuracy when the close-talking microphone used to collect testing data was switched to a desk-top microphone. In this case, the effects of additive noise and linear filtering in the testing environment cause a mismatch between the acoustical features extracted from speech recorded in the training and testing environments. Similar performance degradation has been reported in other adverse environments including automobiles [*e.g.* 30, 52, 15], telephone lines [*e.g.* 76, 58, 13], aircraft cockpits, and noisy work offices [*e.g.* 32, 16, 62].

When a speech recognition system is deployed in the field, it is very common to observe adverse factors that did not exist (or were not observed) in laboratory experiments. Therefore, depending on the specific application, recognition systems must handle environmental mismatches between training and testing conditions to maintain the same level of accuracy. In the sections below we discuss several general approaches to improving the robustness of speech recognition with respect to environmental variability: re-training, multi-style training, and environmental adaptation.

1.1.1. Re-Training

Perhaps the most obvious way to avoid the problem of environmental variability is to simply re-train an "environment-dependent" system with data from new testing environments. This is analogous to the re-training (or enrollment) process for new speakers in a speaker-dependent system. For a speaker-dependent recognition system, a new user must enroll in the system in order to have the recognizer adapted to the new speaker for good performance. By the same token, speech recognizers are usually “environment-dependent” systems and can learn to work in new environments by re-training using training data from the target environment.

However, several barriers exist for the application of re-training approach:

- **Data collection**: New training data sets need to be collected for reliable training. The process of collecting training data itself is a time-consuming process. For example, the training corpus in the ARPA WSJ0 task [83] consists of 12 hours of speech and the collection of such a huge amount of data can take several months.
- **Training time and storage requirements**: The training process is extremely computationally expensive. For small tasks such as an alphanumerical database, the training process can take hours. For large tasks like 20,000-word dictation applications, training may take days or weeks. Furthermore, the storage requirements incurred when training on a large amount of data are also considerable.
- **Lack of *a priori* knowledge**: When a speech recognition system is deployed in the field, it is difficult to predict in advance precisely how the testing environments will change.

1.1.2. Multi-Style Training

Another straightforward way to cope with the environment variability is to train an “environment-independent” system using the approach of multi-style training. This is implemented by pooling data from different acoustical environments, similar to the common strategy for speaker-independent systems, which is to combine training data from a number of speakers.

There are several issues that need to be taken care of in multi-style training.

- **Limited recognition accuracy**: In an alphanumerical task, Acero [1] showed that multi-style training increased the robustness for the cross-microphone experiments at the expense of sacrificing performance with respect to the case of training and testing on the same condition.

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

Table 1-1. Comparison of recognition accuracy obtained using multi-style training and two environments, the Sennheiser close-talking microphone (CLSTK), and the desktop Crown PZM (CRPZM) as reported by Acero [1].

Lippmann [62] reported a similar degradation in performance in a task of different speech style. This degradation probably occurs because some fine acoustic properties observed in one environment are blurred by blending different training environments.

- **Lack of sufficient environments**: To achieve microphone-independence using multi-style training, data from various microphones and acoustical environments will be necessary. The problem is the number of environments. For the similar problem of speaker variability, Lee and Hon [55] found that 80 speakers were needed to achieve speaker independence. It is not clear how many different acoustical environments would be necessary to provide sufficiently broad coverage to obtain microphone independence.

1.1.3. Environmental Compensation Using Dynamic Adaptation

Although re-training and multi-style training are possible solutions to environment independence, both of these approaches exhibit the problems described above. It would be more useful and desirable for the system to learn the characteristics of the target testing environment and adapt accordingly. In this dissertation, we have chosen to address the problem of environmental variability based on environmental adaptation. As with the speaker adaptation for speaker-independent recognition, it should be possible to improve speech recognition accuracy by adapting to the target environment rapidly and non-intrusively.

When changes of environment are encountered, possible adaptation strategies include: (1) modification of the features extracted by the recognition system to provide a better characterization of the acoustical properties of the target environment, (2) transformation or mapping of the corresponding stored templates containing representations of the spectra in the target environment. In principle (and with a sufficiently large amount of training data), there is no apparent difference between these two approaches except some possible implementation issues. As we can see later in this thesis, these two approaches produce similar results.

1.2. Towards Environment-Independent Recognition

The goals of this thesis are to address the issue of severe performance degradation of speech recognition systems when the acoustical characteristics of the training and testing environments are different, to obtain a better overall understanding of the issue of environmental mismatch, to evaluate the usefulness of several practical compensation techniques, and, finally, to show that environmental robustness can be achieved under a number of acoustical conditions.

Because of the mismatches between training and testing conditions, speech recognition systems usually require re-training for every new condition to achieve high accuracy. On one hand, re-training provides a simple way to alleviate the problem of environmental variability due to mismatches. On the other hand, because some information is contaminated or lost due to noise, environments with a higher level of noise suffer a loss of accuracy.

As part of an investigation of the degradation encountered by different baseline speech recognition systems, it was found [63] that using re-training, recognition accuracy for test data from a desktop microphone was 76.2%, while a recognition accuracy of 78.6% could be obtained from a similar system trained with a close-talking microphone using environmental compensation techniques. This supports our contention that environmental compensation can provide better recognition accuracy than re-training.

1.2.1. Sources of Environmental Variability

There are various types of acoustical degradation that an environment-dependent system can encounter when it is deployed in the field. In this dissertation we develop environmental adaptation algorithms that achieve environment independence for speech recognition. We evaluate the success of these algorithms by comparing their performance using speech that is recorded simultaneously from different microphones (and hence in different environments). In such a database the only variable is the change of microphone, as other factors such as speaker variability are nonexistent.

It is worth noting that the effects of microphone change can include: (1) the introduction of unknown linear filtering due to different transducer characteristics, (2) changes of additive ambient noise due to differences in the noise-cancellation properties of the microphones, (3) different levels of room reverberation due to different mounting positions, and (4) different levels of competing interference due to different directivity of microphones.

1.2.2. Performance Evaluation

Modern speech recognition systems are normally evaluated by observing the recognition produced by them using a standardized corpus of speech material for testing and training. In this thesis we primarily use a standard corpus designed for the ARPA spoken language community, in which sentences are read from articles in the Wall Street Journal [67]. For most of the experiments we use a subset of this database that provides speaker-independent speech with a closed vocabulary of

5,000 words. Speech in this database is recorded simultaneously using two microphones: (1) the standard Sennheiser HMD-414 (or HMD-410 for some recordings) and (2) an unknown member of a set of 15 alternate microphones (referred to as "secondary microphones" in this thesis).

In general the speech recognition systems used in our experiments are trained using the standard ARPA close-talking microphone. The various compensation algorithms are normally applied to the testing material during the recognition phase.

1.3. Dissertation Outline

Chapter 2 provides an overview on the fundamentals of environmental robustness of speech recognition. It describes major sources for performance degradation a recognition system encounters in real-world application. It also reviews previous algorithms in related work.

Chapter 3 contains a brief description of the CMU SPHINX-II speech recognition system, and describes the tasks and databases used in our evaluations.

Chapter 4 describes and evaluates the Blind SNR-Dependent Cepstral Normalization (BSDCN) algorithm to compensate for environmental mismatches. BSDCN employs additive correction vectors that depend exclusively on the instantaneous SNR of the input signal. These vectors are obtained in an environment-independent fashion by establishing correspondence between SNR histograms representing the training and testing environments.

Chapter 5 describes the Multiple Fixed Codeword-Dependent Cepstral Normalization (MFCDCN) and Interpolated Multiple Fixed Codeword-Dependent Cepstral Normalization (IMFCDCN) algorithms to achieve environment independence. The incorporation of environment-specific information learned from pre-existing speech frames is also discussed in the context of MFCDCN and IMFCDCN.

Chapter 6 presents the Phone-Dependent Cepstral Normalization (PDCN) algorithm and its extensions. PDCN compensates for environmental variabilities based on the presumed phonetic identity of a speech segment as revealed by the search process. Further experiments are carried out in an effort to combine PDCN with MFCDCN.

Chapter 7 describes the use of codebook adaptation techniques for environmental adaptation. Two kinds of approaches, the Dual-Channel Codebook Adaptation (DCCA) and Baum-Welch

Codebook Adaptation (BWCA) algorithms, are proposed and evaluated. These techniques modify the stored reference templates of the recognition system, rather than the feature vector that is input to the system.

Chapter 8 describes the contributions of our work and provides some suggestions for future work.

Chapter 2
Overview of Environmental Robustness in Speech Recognition

In this chapter we will describe and discuss several sources for the degradation of recognition accuracy in the context of changes of acoustical environments. We will also describe several approaches that have been described previously in related work. Finally, we briefly summarize the advantages and disadvantages of the various approaches.

2.1. Sources of Degradation

In this section we describe various sources of acoustical degradation in adverse environments. The sources include additive noise, linear filtering and other sources such as articulation effects induced by environmental influence [52, 87], transient or impulse noise, noise due to transmission and switching equipment in a telephone network, and interference by speech signals from other speakers talking simultaneously.

2.1.1. Stationary Noise

The performance of speech recognition systems degrades drastically when training and testing are carried out with different noise levels. For example, it was found [70] that with zero-mean, white Gaussian noise added to an utterance at each specified SNR, the accuracy of speech recognition can drop from 95.6% when clean speech is used, to 46.3% when noisy speech at a 15-dB global SNR is used.

Various types of noise can be found in a passenger car [30, 52, 15], such as running noise from engine and tires, "function" noise from the car radio and windshield wipers, and non-stationary noise from other passing vehicles. It is not unusual [52, 15] that the global signal-to-noise ratio of speech signal recorded in a passenger car can drop below 5 dB when the car is cruising at speeds of 90 km/h. To achieve high accuracy in presence of these noises is a difficult task. Another example is speech recognition in the cockpit of a modern jet fighter aircraft, in which the high level of noise presents an extremely challenging problem to automatic speech recognition. Even in an office environment, ambient noise can have a major impact on speech recognition accuracy.

Background ambient noise is usually considered to be additive noise that can be modeled as a stationary random process that is uncorrelated with the speech signal. A great amount of work has been carried out in an attempt to minimize the effect of this corrupting noise. Many early speech enhancement techniques developed to combat the effects of additive noise have been summarized by Lim [60]; later approaches and their results were reviewed by Juang [45]. These techniques have achieved some improvements in accuracy to differing extents.

2.1.2. Linear Filtering

In addition to additive corruption by noise-like signals, speech may undergo a series of spectral distortions while being produced, recorded, and processed for recognition. For example, walls and other obstacles in the room where the speech recognizer is deployed can produce multiple reflections which influence the signal spectrum. Depending on its type and location, the recording microphone can also have significant impact on the speech spectrum. When the microphone used to collect testing data is different from that used to collect training data for building the reference patterns, a mismatch in average spectrum is produced, which becomes a major problem for speech recognition [1, 99].

Telephone channels provide an additional source of distortion by linear filtering. Channel-induced variation on telephone lines can also be regarded as a combination of additive noise and channel filtering. It has been found [58] that the error rate of a speech recognizer can increase from 1.3% to 44.6% when the testing data are filtered by a pole/zero filter modeling a typical long-distance telephone line and corrupted by noise at a global SNR of 15 dB. Similarly, it was also reported [76] that in an alphanumerical task, the recognition accuracy of the CMU SPHINX-II system dropped from 82.8% to 68.0% when the testing utterances were degraded by using a telephone network simulator to match the frequency response of the CCITT 1025 channel [26].

2.1.3. Other Factors

2.1.3.1. Articulation Effects

In adverse environments, a talker can consciously or unconsciously change his/her speaking manner in order to overcome changes in ambient acoustical conditions. The variance of speaking rate and the psychological status from the speaker also affect the acoustical characteristics like sound formats and rhythmic stability [52]. Even differences in the physiology of the vocal tract will cause variability in the production of speech signals.

One well-known problem is the Lombard effect, which refers to changes in articulation caused by the presence of high-intensity noise. It has been also reported [62] that these changes in articulation can degrade the performance of a speech recognizer considerably.

2.1.3.2. Transients

In practical applications of speech recognition, the speech signals can be corrupted by high-intensity transient noises produced by door slams, phone rings, or other similar sources [106]. Other transient noises can include noise from passing motor vehicles in automotive applications [30], random start-up and shut-off of machinery in a factory environment, etc. In some cases the amplitude of the transients is sufficient to mask the target speech signal. These high-intensity transient noise sources represent a difficult task for speech recognition.

2.1.3.3. Transmission and Switching Noise

Several types of interference or noise can arise in the transmission of signals over telephone lines, including amplitude jitter, phase jitter, and additive low-frequency tones [11, 76]. Quantization error may also be present when certain speech coding techniques are in use. For applications in cellular telephony, passing from one cell to another cell can introduce switching noise.

2.1.3.4. Interference from Other Speakers

One of the most challenging problems for robust speech recognition is interference from other speakers talking simultaneously (the cocktail party effect). Until very recently, most recognition systems modelled the incoming speech using linear prediction, which assumes a time-varying all-pole filter excited by either an impulse train or white noise. Since this model assumes that a single voice is present, interference from other speakers will cause a dramatic degradation in recognition accuracy. Furthermore, because other background interference like music or talk-radio is highly non-stationary and frequently speech-like, the co-channel problem makes accurate speech recognition difficult to achieve when the SNR is very low [65].

2.2. Review of Previous Related Work

In the previous section, we described a few major sources of performance degradation that a speech recognition system may encounter when it is deployed in an acoustically adverse environment. A number of signal processing approaches have been proposed to improve acoustical robustness in speech recognition systems:

- The application of array-processing techniques using multiple microphones.
- The use of physiologically-motivated models (auditory models) to approximate the environmental robustness of the human auditory system.
- The approach of signal decomposition using hidden Markov Models.
- The use of noise-word modeling.
- The application of highpass filtering of cepstral coefficients.
- The use of various pre-processing techniques to "enhance" speech signals in the presence of additive noise and linear filtering.
- Other techniques including robust distortion measures, adaptive noise cancellation, noise-masking techniques.

In the following sections, we will review these approaches and provide a brief discussion of their capabilities and limitations.

2.2.1. Array Processing Using Multiple Microphones

Several different types of array processing techniques have been applied to speech recognition systems. The simplest array-processing approach is that of the delay-and-sum beamformer [21, 22]. In this method, enhancement of the desired signal is accomplished by applying steering delays to the outputs of the microphones to compensate for differences among the microphones of the arrival times of the direct field signal from the desired source.

Several researchers have used techniques based on minimizing mean square energy using classical adaptive filtering, such as the Frost algorithm [108, 85]. In general, these algorithms are useful in providing greater sensitivity in the direction of the desired signal while providing nulls in the direction of undesired noise sources. On the other hand, they assume that the desired signal is statistically independent of all sources of degradation. Therefore, these algorithms are not effective for applications in reverberant rooms where the distortion includes a delayed version of speech signal.

Multi-microphone correlation-based processing that mimics the processing performed by the human binaural system is also being explored by researchers [101]. However, most studies have employed cross-correlation-based processing to identify the direction of a desired signal, rather than to enhance the quality of the desired input for speech recognition.

2.2.2. Auditory-Model-Based Front Ends

Because the human auditory system is very robust to changes of acoustical environment, a number of researchers have proposed various signal processing schemes that mimic the processing of the auditory periphery. In an attempt to duplicate the human auditory capability to combat the effect of noise and/or unknown linear filtering, researchers have proposed models of the peripheral auditory system with different emphasis on various aspects of the physiology.

Ghitza [27] proposed a computational model to simulate the auditory-nerve firing pattern that may be robust to noise corruption. It was applied to a task of digit-recognition in noise and showed improvement over a conventional front-end. Meng [73] showed that the physiologically-motivated model of Seneff [97] improved the performance in comparison to several types of conventional signal processing approaches in both clean and noisy environments. Other applications of auditory-model-based approaches include physiologically-motivated models by Ohshima [81] and cochlear models by Lyon [68].

Another simple front end loosely motivated by the human auditory system is the use of Mel-frequency cepstrum [17]. Mel-frequency cepstrum is simple and computationally inexpensive whereas physiologically-motivation models are computationally expensive.

Though most of these techniques do indeed improve the robustness of speech recognition system under certain conditions, their computational cost is very high when compared with more conventional front ends. Furthermore, most of the experimental comparisons used to evaluate the models have been limited to the effects of degradation introduced by artificially-added white noise. Realistic acoustical environments, on the other hand, introduce a number of other types of degradation as well, as noted above.

2.2.3. Signal Decomposition using Hidden Markov Models

Varga and Moore [103], and Gales and Young [24] proposed an approach called “signal decomposition using hidden Markov models” in which separate HMMs are developed to characterize the desired speech and simultaneous noise. The decomposition of noise and speech was implemented using a three-dimensional Viterbi search with the goal of accounting for the combined observation of speech and noise. Unlike the approach of noise-word modeling in which noise is assumed to occur between speech segments, noise and speech are assumed to occur simultaneously in this approach.

While the approach of simultaneous decomposition is powerful, the computational cost is extremely high because the Viterbi search must be performed in three dimensions. Furthermore, this approach lacks the ability to compensate for unknown linear filtering.

2.2.4. Use of Noise-Word Models

The use of noise models has been explored in some studies to characterize some non-verbal events that are often observed in spontaneous speech. The goal of this approach is to identify the non-speech sound by deriving hidden Markov Models (HMMs) for these "noise" words. For example, Ward [106] applied a technique of "non-verbal sound modeling" to model the effects of transient noise such as breath noises, lip smacks, filled pauses, paper rustles, telephone rings, door slams, and other non-stationary noise. A similar "noise-word modeling" approach has been proposed by Wilpon [109] using statistical models for extraneous speech and background.

While noise-word modeling is useful for some transient noise, it does not take into account non-stationary noise during speech segments. A database of speech with labeled noise segments is needed to identify the noise words in this approach. This approach is not able to handle the effect of linear filtering, nor can it adequately cope with noise that occurs during the speech sounds.

2.2.5. High-Pass Filtering of Cepstral Coefficients

In dealing with changes of acoustical environment, bandpass or highpass filtering of the cepstral coefficients is useful in reducing slow-varying channel effects. RASTA (RelAtive SpecTrA Processing) [34] and cepstral mean normalization [64] are techniques that suppress slow-varying channel effects in each log spectral or cepstral component. In the RASTA method, a bandpass or highpass filter with a very low cutoff frequency is applied to the running estimate of each log spectral or cepstral component. Cepstral mean normalization subtracts the mean of cepstral vectors from the cepstral coefficients of an entire utterance on a sentence-by-sentence basis.

The application of RASTA can usually reduce the channel mismatch between training and testing conditions [99,100]. However, RASTA might not be helpful in situations where no mismatch is present between these conditions. In our pilot experiment [46], we found that RASTA provided a moderate improvement for mismatched conditions at the price of a performance degradation (11% relative error increase) for matched conditions.

On the other hand, cepstral mean normalization (CMN) improves the robustness of the recognizer even when the training and testing conditions are the same. The performance difference between CMN and RASTA might be due to a possible information loss as the filtering operation in RASTA removes not only the constant component but also some other slow changes in each cepstral component.

Because CMN is simple and effective, we include it as a standard operation throughout this thesis unless otherwise specified.

2.2.6. Codeword-Dependent Cepstral Normalization (CDCN)

Acero [1] proposed the codeword-dependent cepstral normalization algorithm (CDCN) as a pre-processing technique that can eliminate the effects of linear filtering and additive noise. CDCN uses EM techniques to compute maximum likelihood (ML) estimates of the environmental parameters that characterize the contributions of additive noise and linear filtering. These environmental parameters are chosen to best match (in the ML sense) the ensemble of cepstral vectors of the incoming speech to the ensemble of cepstral vectors in a universal codebook which is generated from the training corpus. The estimate of each clean speech cepstral vector is obtained using a MMSE estimator based on information from the environmental parameters.

Using CDCN, recognition accuracy when training on the standard close-talking (CLSTK) microphone and testing on the desk-top (PZM6FS) microphone can improve to the level observed when the system is both trained and tested on the PZM6FS. The CDCN algorithm has the advantage that it does not require *a priori* knowledge about the testing environment, but it is more computationally demanding.

To facilitate computational simplification, CDCN makes some assumptions that are valid for high-SNR signals but not valid for low-SNR signals. Although it is quite effective for high-SNR signals, CDCN is not as effective for low-SNR signals. The use of a universal codebook that characterizes a universal space with only a limited codewords can cause degradation for testing data from the training environment due to the limited dimensionality. Because conventional CDCN must apply the normalization process to the training data as well as to the testing data, the typical speech recognition system must be completely retrained to incorporate the CDCN algorithm.

2.2.7. Environment-Specific Cepstral Normalization

Acero [1,2] also proposed several pre-processing techniques that compensate for environmental mismatches by transforming noisy testing speech to the acoustical space of the training environment in an environment-specific manner. Two of these environment-specific algorithms are the SNR-Dependent Cepstral Normalization (SDCN) and Fixed Codeword-Dependent Cepstral Normalization (FCDCN) algorithms.

SNR-dependent cepstral normalization applies an additive correction in the cepstral domain, with the compensation vector depending exclusively on the instantaneous SNR of the signal. The compensation vectors equal the difference of the average cepstra between simultaneous stereo recordings of speech signal from both the training and testing environments for each SNR of speech. At high SNRs, this compensation vector primarily compensates for differences in spectral tilt between the training and the testing environments, while at low SNRs the compensation vector provides a form of noise substraction.

The fixed codeword-dependent cepstral normalization algorithm (FCDCN) combines some of the more attractive features of the CDCN and SDCN algorithms. Like SDCN, the compensation factor equals the difference in cepstra between the training and testing environments, but like CDCN, the compensation factor is different for different VQ codewords as well. This algorithm provides more detailed compensation which makes use of the SNR and VQ codewords. It is also simple and efficient, and can achieve a level of recognition accuracy comparable to that of CDCN.

The SDCN and FCDCN algorithms are computationally simple and effective. Both SDCN and FCDCN are based on a data-driven approach, in contrast to CDCN which accomplishes normalization using a structural model of acoustical degradation. Furthermore, the compensation provided by SDCN and FCDCN is applied only to speech in the testing phase, but not in training the system. Hence, the use of SDCN and FCDCN does not involve a tedious re-training process.

A major disadvantage of both SDCN and FCDCN, however, is that their compensation vectors must be recomputed for each new testing environment. This recomputation of the compensation vectors requires a training database which contains simultaneously-recorded speech samples in the training and testing environments. In many applications, such a database is unavailable.

2.2.8. Adaptive Labeling

Nadas et al [78] proposed the *adaptive labeling* algorithm in an attempt to diminish the degradation of performance that occurs as a result of changes in the signal characteristics following changes in ambient noise and other recording environment conditions. The technique is based on the use of a codebook of prototype vectors obtained from the "clean" training data computed under the assumption that the distribution of the codebook reflects the distribution of the training data.

Vector quantization is used as a tool in the adaptive labeling approach for defining adaptive transformations for the normalization of speech. The noisy input is converted into normalized output so that the distribution of the normalized observation is as similar as possible to the distribution of the training data characterized by the available prototype. In addition, the transformation used in adaptive labeling is itself continuously updated from spectral vector to spectral vector. This update is accomplished by perturbing the current transformation to generate a new transformation that reduces the error between the current transformed output and the corresponding closest centroid in the codebook.

When applied to a speaker-dependent 5000-word speech recognition system, adaptive labeling achieved an average of 80% error rate reduction for situations where degradation was caused by a variations in distance between talker and microphone, loudness of the speech, or movement by the talker [78].

Adaptive labeling is similar to the SDCN and FCDCN algorithms in some ways in that it aims to produce a sequence of spectral vectors which has the same mixture distribution as the training data. On the other hand, adaptive labeling employs a continuous update of transformation for the next spectral vector. Relying on the closest prototype makes the adaptive label algorithm a decision-directed algorithm [108]. Decision-directed algorithms generally produce stable performance as long as the initialization of the parameters of the system provide a sufficient level of accuracy for the decision being made. However, the distribution of the adapted process may, in theory, fail to converge to the true distribution.

2.2.9. Other approaches

Many other approaches and algorithms have also been proposed to render practical speech recognition systems more robust in the presence of conditions like ambient noise, distortion, room echoes, and other noised-induced problems. These include the use of robust distortion measures [70, 12], adaptive noise cancellation using two signal sources [107, 30], noise cancellation using estimate-maximize procedures with a single-channel input [88, 82, 18], the use of noise-cancelling microphones [15, 104], the use of room modeling for echo cancellation [31], and noise-masking schemes [47, 102].

2.2.10. Discussion

In this section we have reviewed several techniques to cope with the issue of acoustical mismatches between the training and testing environments. Although these techniques have provided useful improvements in performance in some applications, many of them are not appropriate in the SPHINX-II system.

The major goal of this dissertation has been to develop algorithms that can enable the speech recognition system to adapt to new environments with high recognition accuracy and with low computational complexity, but without environment-specific re-training.

One major factor in our selection of techniques to be considered is that of computational complexity. A candidate compensation algorithm must be simple enough to be applied to many utterances over and over again to a speech recognition system that is already quite complex. We elect not to use the approaches of auditory models or the decomposition of noisy speech using parallel HMMs because of possibly limited or no benefit at a high computational cost. Since we are concerned with environments in which signals are corrupted by both linear filtering and additive noise, the techniques of noise masking, adaptive labeling, and noise-word modelling are not appropriate for our problem.

We have also avoided the use of CDCN in our work, in part because of our concerns about the inadequacy of its assumptions for speech at low SNRs and in part because approaches of higher accuracy and lower computational complexity in testing are desired. The present SDCN and FCDCN algorithms are also inappropriate for our work in their present form because of their reliance on the existence of “stereo” training data sets when new environments are considered.

We are concerned with degraded speech recorded using a single microphone in this work, so we do not consider multi-microphone approaches such as microphone array processing and adaptive noise cancellation using two sensors.

We do make routine use of the cepstral normalization technique (CMN) because of its simplicity, and because it outperformed the RASTA method in pilot studies in our environment [64].

2.3. Summary

In this chapter, we described and identified several sources for degradation of recognition accuracy in the context of changes of acoustical environments. Various adverse components present in the issue of environmental variability have been discussed, including additive ambient noise, unknown linear filtering, impulsive noise, Lombard effect, competing interference from other speakers, and so on.

We also reviewed most of the techniques that have already been proposed as potential solutions for the problem of robust speech recognition in adverse environments, although many of these techniques do not provide useful solutions to our short-term research needs.

Our original research will focus on the development of signal processing algorithms of moderate complexity which can provide substantial improvements in environmental robustness, and which do not require any *a priori* information about unknown testing environments.

Chapter 3
The SPHINX-II Recognition System

Since the environmental adaptation algorithms to be developed will be evaluated in the context of continuous speech recognition, this chapter will provide an overview of the basic structure of the recognition system used for the evaluations. Because the development of CMU speech recognition system is a continuously dynamic effort, some system changes have taken place during the development of the algorithms. In order to make fair comparisons, SPHINX-II is used as the platform to evaluate these approaches whenever possible, regardless of whether some of the algorithms were originally developed before SPHINX-II was developed. Although we use the SPHINX-II system to develop and evaluate our algorithms, they can also be easily applied to other recognition systems.

The most important topic of this chapter is a description of various aspects of the SPHINX-II recognition system. We also summarize the data collection and system evaluation procedures used in the thesis.

3.1. An Overview of the SPHINX-II System

SPHINX-II is a large-vocabulary, speaker-independent, Hidden Markov Model (HMM)-based continuous speech recognition system, like its predecessor, the original SPHINX system (which is now referred to as SPHINX-I). SPHINX was developed at CMU in 1988 [101,54] and was one of the first systems to demonstrate the feasibility of accurate, speaker-independent, large-vocabulary continuous speech recognition. The performance of the recognition system has been significantly improved by a number of modifications and changes. The new system has been renamed SPHINX-II [40, 42] in order to distinguish it from the original SPHINX system.

We first summarize some of the major differences between SPHINX-II and SPHINX. Further details of the SPHINX system can be found in the literature [*e.g.* 55, 56,57]. A brief description of the overall structure of the SPHINX-II system will then be provided.

The major differences between SPHINX and SPHINX-II are:

- SPHINX-II uses Mel-scale frequency cepstral coefficients (MFCC) [40] as its feature set; SPHINX uses frequency-warped linear-prediction-based cepstral coefficients (LPCC) as its features.

- SPHINX-II is based on semi-continuous hidden Markov models while SPHINX uses the approach of discrete-density hidden Markov models [96]. Specifically, the output distributions used in SPHINX-II are weighted mixtures of the best 4 Gaussian distributions out of 256, while the output distributions in SPHINX are discrete.
- SPHINX-II uses subphonetic shared-distribution models [42] to provide further acoustical modeling between different acoustic-phonetic phenomena. In contrast, SPHINX uses generalized triphones to achieve acoustical modeling.
- A three-pass search algorithm [3] is employed in SPHINX-II to process very large vocabulary and long-distance language models efficiently while the original SPHINX system uses the one-pass Viterbi beam search [105, 67, 56].

Figure 3-1 shows the fundamental structure of the SPHINX-II system. We describe the functions of each block briefly and make a comparison among the differences between SPHINX and SPHINX-II as necessary.

3.1.1. Signal Processing

Almost all speech recognition systems use a parametric representation of speech rather than the waveform itself as the basis for pattern recognition. The parameters usually carry the information about the short-time spectrum of the signal. SPHINX-II uses mel-frequency cepstral coefficients (MFCC) as the static features for speech recognition [40]. First-order and second-order time derivatives of the cepstral coefficients are then derived, and power information is included as a fourth feature. The front end of SPHINX-II is illustrated in Figure 3-2. We summarize this feature extraction procedure as follows:

1. The input speech signal is digitized at a sampling rate of 16 KHZ.
2. A pre-emphasis filter $H(z) = 1 - 0.97z^{-1}$ is applied to the speech samples. The pre-emphasis is used to reduce the effects of the glottal pulses and radiation impedance [71] and to focus on the spectral properties of the vocal tract.
3. Hamming windows of 25.6-ms duration are applied to the pre-emphasized speech samples at an analysis rate (frame rate) of 100 windows/sec.
4. The power spectrum of the windowed signal in each frame is computed using a 512-point

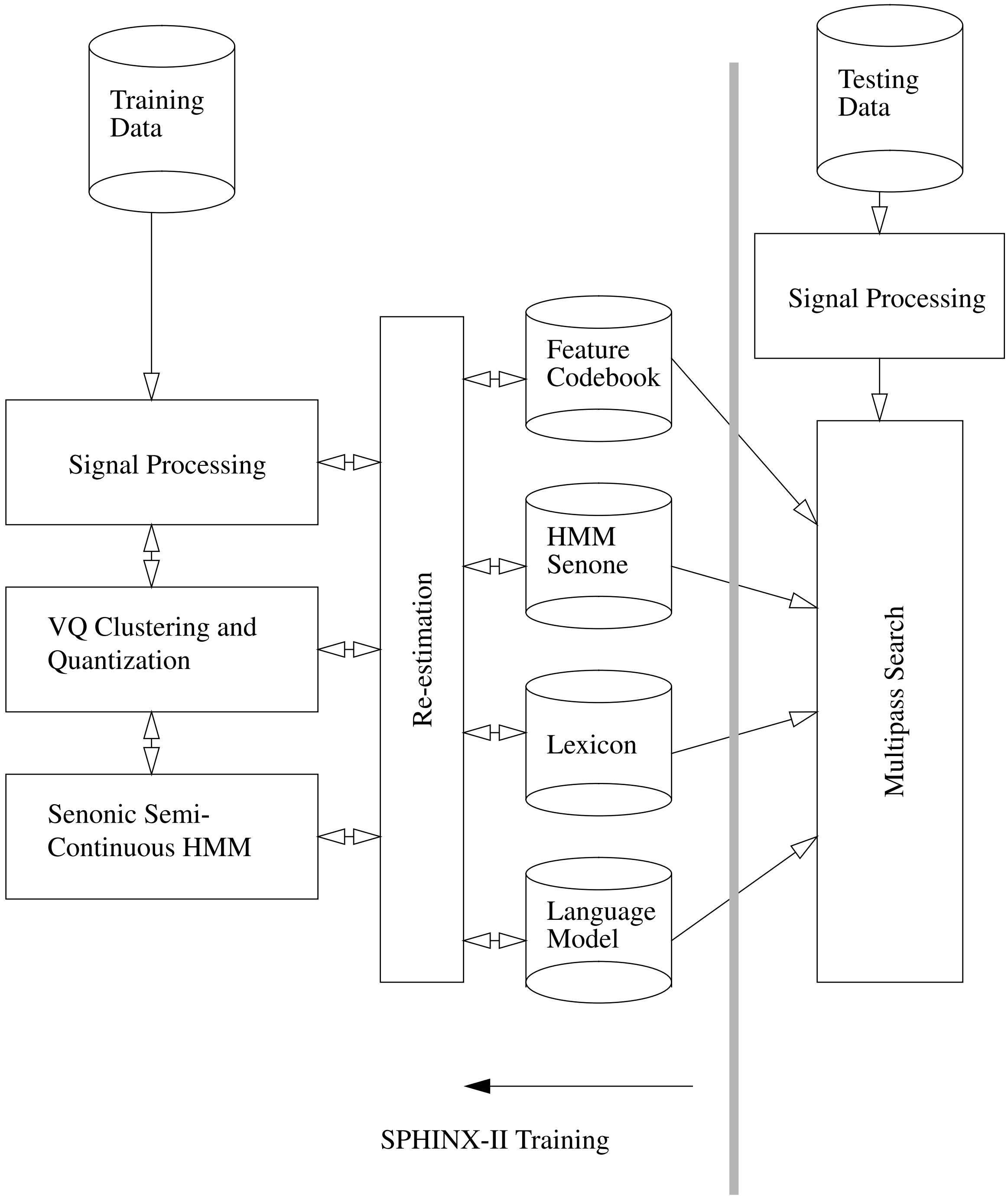


Figure 3-1. Block diagram of SPHINX-II.

DFT.

5. 40 mel-frequency spectral coefficients (MFSC) are derived based on mel-frequency bandpass filters with 13 linear bands for 100 Hz to 1 kHz and 27 logarithmic bands for 1 kHz to 7 kHz.

6. For each 10-ms time frame, 12 mel-frequency cepstral coefficients (MFCCs) are computed using the cosine transform as shown in Equation (3.1)

$$\boldsymbol{x}_t[k] = \sum_{i=1}^{40} X_{t,i} \cos\left[k(i-1/2)(\pi/40)\right], \quad 1 \le k \le 12 \tag{3.1}$$

where $X_{t,i}$, represents the log-energy output of the i^{th} mel-frequency bandpass filter at time frame t.

7. The derivative features are computed from the static MFCCs as follows,

(a) Differenced cepstral vectors consist of 40-ms and 80-ms differences with 24 coefficients.

$$\Delta\boldsymbol{x}_t(k) = \boldsymbol{x}_{t+2}(k) - \boldsymbol{x}_{t-2}(k), \quad 1 \le k \le 12$$

$$\Delta\boldsymbol{x}'_t(k) = \boldsymbol{x}_{t+4}(k) - \boldsymbol{x}_{t-4}(k), \quad 1 \le k \le 12$$

(b) Second-order differenced MFCCs are then derived in similar fashion, with 12 dimensions.

$$\Delta\Delta\boldsymbol{x}_t(k) = \Delta\boldsymbol{x}_{t+1}(k) - \Delta\boldsymbol{x}_{t-1}(k), \quad 1 \le k \le 12$$

(c) Power features consist of normalized power, differenced power and second-order differenced power.

$$\bar{\boldsymbol{x}}_t(0) = \boldsymbol{x}_t(0) - \max_i \{\boldsymbol{x}_i(0)\}$$

$$\Delta\boldsymbol{x}_t(0) = \boldsymbol{x}_{t+2}(0) - \boldsymbol{x}_{t-2}(0)$$

$$\Delta\Delta\boldsymbol{x}_t(0) = \Delta\boldsymbol{x}_{t+1}(0) - \Delta\boldsymbol{x}_{t-1}(0)$$

In summary, SPHINX-II uses the Fourier-spectrum-based MFCC as the parametric representation for a signal in contrast to the LPC-derived cepstra used in the original SPHINX system. We found that SPHINX-II achieved about 10% word error rate reduction using MFCC in the context of February 1992 dry-run test set from Wall Street Journal CSR task compared to LPCC.

Thus, the speech representation uses 4 sets of features including: (1) 12 Mel-frequency cepstral coefficients (MFCC); (2) 12 40-ms differenced MFCC and 12 80-ms differenced MFCC; (3) 12 second-order differenced cepstral vectors; and (4) power, 40-ms differenced power, and second-order differenced power. As in the original SPHINX, these features are all assumed to be statistically independent for mathematical and implementation simplicity.

3.1.2. Vector Quantization

Discrete HMMs are often used as the initial models to train semi-continuous HMMs. For discrete HMMs, each time frame of speech is represented by a symbol rather than a continuous vector $\boldsymbol{x}_t$. Hence speech samples must be transformed into a sequence of symbols during the training and testing phases.

Vector quantization (VQ) [29,61], is a data-reduction technique in which a feature vector is mapped into a discrete symbol. A vector quantizer is defined by a codebook which consists of a set of prototype vectors, and a distortion measure that estimates the similarity of two vectors. The distortion measure used for VQ in SPHINX and SPHINX-II[1] is the Euclidean distance.

As described earlier, SPHINX-II uses four sets of features as parametric representation for each time frame of speech. One codebook of 256 prototype vectors, or codewords, is generated for each parametric representation of speech, using approximately 10^5 to 10^6 feature vectors. These prototype vectors represent the distributions of the training vectors in each feature space. They are estimated using a hierarchical clustering algorithm [61], similar to the K-means algorithm, which builds an entire codebook by spliting each existing cluster into two smaller clusters. A prototype vector in each codebook is a centroid of a cluster with similar feature vectors.

As mentioned above, each of the four feature codebooks are assumed to be independent for simplicity. Each cluster in these codebooks is modeled by a Gaussian distribution with a diagonal covariance matrix. The covariance matrix is assumed to be diagonal for reduction of computation and for a reliable estimation with a limited amount of training data. In addition to the mean vectors, the covariance matrix of each cluster is computed to initialize the training of the SC-HMM codebook.

1. VQ with Euclidean distance metrics is only used at initialization time.

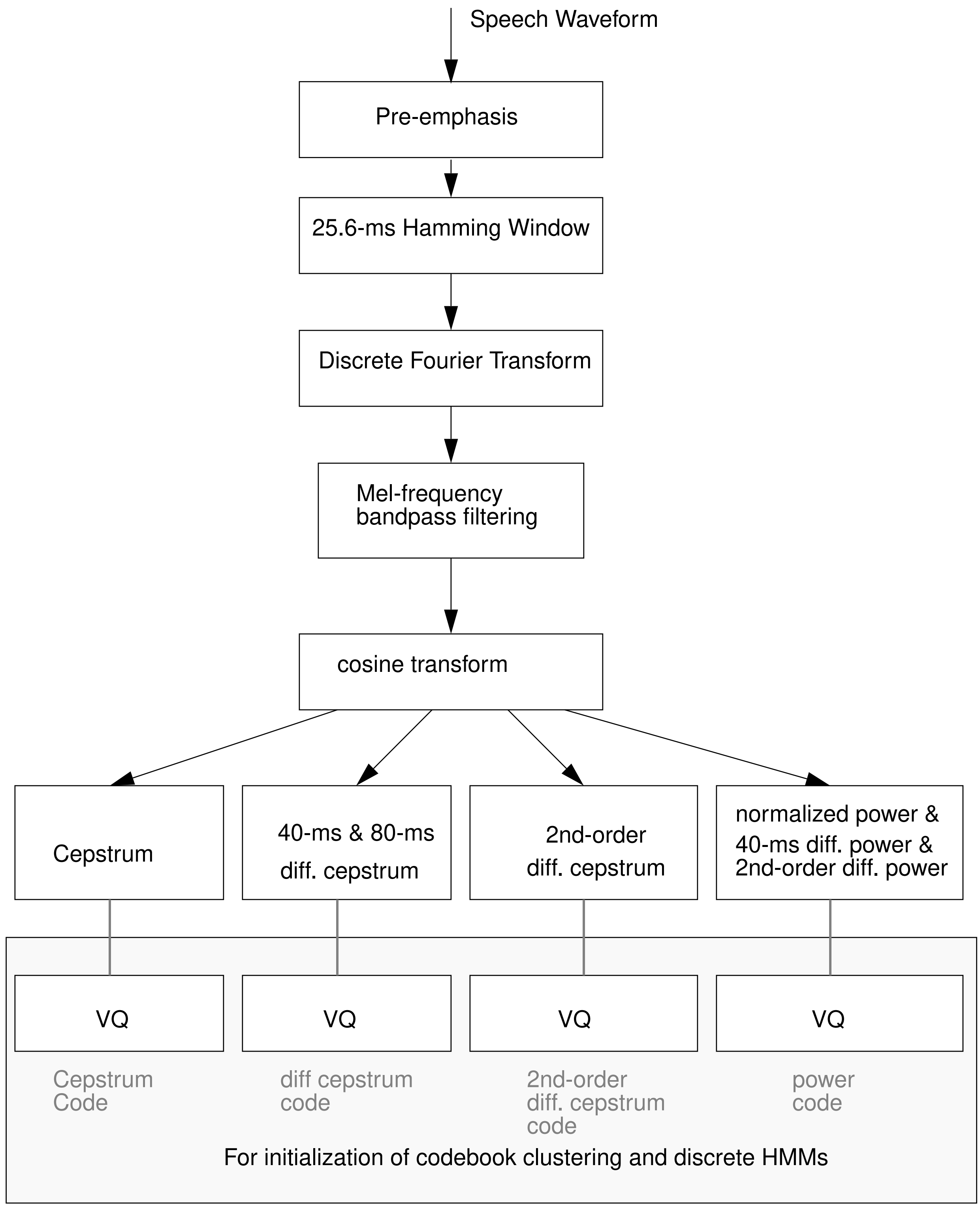


Figure 3-2. Block diagram of SPHINX-II's front end.

For discrete HMMs (D-HMMs), every speech input vector is mapped to four 1-byte representations, one for each feature set, after vector quantization. In contrast, semi-continuous HMMs (SC-HMMs) need 52 (=13x4) bytes to represent every speech vector (twelve floating-point MFCCs and one power coefficient). So the storage requirement for each speech frame in a D-HMM is 1/13 of that in a SC-HMM but at the expense of vector quantization errors. The context-independent phonetic D-HMM will be used as initial models for training context-dependent SC-HMMs.

3.1.3. Hidden Markov Models

In the context of statistical methods for speech recognition, hidden Markov models (HMMs) have become a well known and widely used statistical approach to characterizing the spectral properties of frames of speech. As a stochastic modeling tool, HMMs have an advantage of providing a natural and highly reliable way of recognizing speech for a wide variety of applications. Since the HMM also integrates well into systems incorporating information about both acoustics and semantics, it is currently the predominant approach for speech recognition. We present here a brief summary of the fundamentals of HMMs. More details about the fundamentals of HMMs can be found in [5, 44, 6, 59, 89].

Hidden Markov models are a "doubly stochastic process" in which the observed data are viewed as the result of having passed the true (hidden) process through a function that produces the second process (observed). The hidden process consists of a collection of states (which are presumed abstractly to correspond to states of the speech production process) connected by transitions. Each transition is described by two sets of probabilities:

- A **transition probability**, which provides the probability of making a transition from one state to another.
- An **output probability** density function, which defines the conditional probability of observing a speech feature when a particular transition takes place. For discrete HMMs (as in the original SPHINX), it is assumed that the observed speech signal is a symbol from a finite alphabet, coded using vector quantization. The output probability function in D-HMMs is modeled explicitly. For semi-continuous HMMs (as in SPHINX-II) or fully continuous HMMs [102], pre-defined continuous distribution functions are used for observations that are multi-dimensional vectors. The continuous density function most frequently used for this purpose is the multivariate Gaussian mixture density function.

The goal of the decoding (or recognition) process in HMMs is to determine a sequence of (hidden) states (or transitions) that the observed signal has gone through. The second goal is to define the likelihood of observing that particular event given a state determined in the first process. Given the definition of hidden Markov models, there are three problems of interest:

- **The Evaluation Problem**: Given a model and a sequence of observations, what is the probability that the model generated the observations? This solution can be found using the forward-backward algorithm [7,89].
- **The Decoding Problem**: Given a model and a sequence of observations, what is the most likely state sequence in the model that produced the observation? This solution can be found using the Viterbi algorithm [105].
- **The Learning Problem**: Given a model and a sequence of observations, what should the model's parameters be so that it has the maximum probability of generating the observations? This solution can be found using the Baum-Welch algorithm (or the forward-backward algorithm) [7, 4].

3.1.4. Recognition Unit

An HMM can be used to model a specific unit of speech. The specific unit of speech can be a word, a subword unit, or a complete sentence or paragraph. In large-vocabulary systems, HMMs are usually used to model subword units [5, 55, 14, 53] such as phonemes, while in small-vocabulary systems HMMs tend to be used to model the words themselves.

Both SPHINX and SPHINX-II are based on phonetic models because the amount of training data and storage required for word models is enormous. In addition, phonetic models are easily trainable. There are 63 basic phones used in the current SPHINX-II system. They include fifty lexical phones, three silence models for silences in different parts of an utterance (beginning, middle and ending part), and ten noise models for non-speech sounds like door slams (pumps), tongue clicks, breath noise, and so on. The phone labels for the basic phone set, along with examples, are listed in Table A-1 in Appendix A.

The phone model is inadequate to capture the variability of acoustical behavior for a given phoneme in different contexts. In order to enable detailed modeling of these co-articulation effects, triphone models were proposed [94] to account for the influence by the neighboring contexts.

Although triphone modeling can account for the left and right phonetic contexts by creating a different model for each possible context pair, it is not actually used directly because the number of triphones is huge. In addition, triphone modeling does not take into account the similarity of certain phones in their effect on neighboring phones. A parameter-sharing technique called distribution sharing [41] is used to describe the context-dependent characteristics for the same phones while triphone generalization [56] was used in SPHINX for the same purpose. The main advantages of these parameter-sharing techniques include: (1) The number of models can be reduced so that the system is more tractable. (2) Parameter-sharing leads to better-trained models with a limited amount of training data.

3.1.5. Training

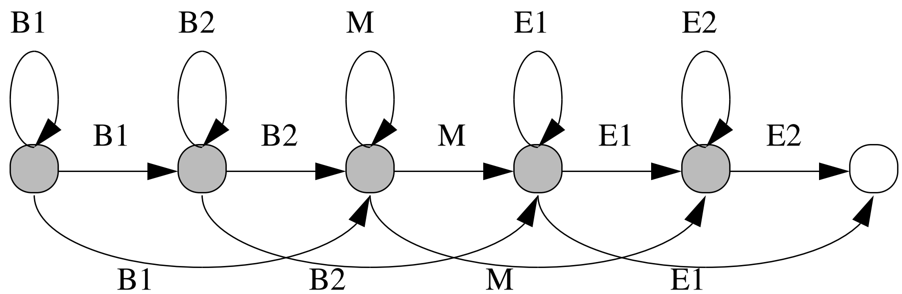


Figure 3-3. The topology of the phonetic HMM used in the SPHINX-II system.

Both SPHINX and SPHINX-II are triphone-based HMM speech recognition systems. Figure 3-3 shows the basic structure of the phonetic model for HMMs used in SPHINX-II. Each phonetic model is a left-to-right Bakis HMM [6] with 5 distinct output distributions. The labeling of the output distribution of each transition is dependent on the source state. The final state in Figure 3-3, which has no outgoing arcs, is added for implementation convenience. Word models are formed by concatenating elementary phonetic models. Sentence models are in turn composed by concatenating component word models.

SPHINX-II [40] uses a subphonetic clustering approach to share parameters among models. The clustering is accomplished at the distribution level instead of at the model level like SPHINX. The output of clustering is a pre-specified number of shared distributions, which are called senones [41]. The senone, then, is a state-related modeling unit. By using subphonetic units for clustering, the distribution-level clustering provides more flexibility in parameter reduction and more accurate acoustic representation than the model-level clustering.

The training procedure involves optimizing HMM parameters given an ensemble of training data. An iterative procedure, the Baum-Welch algorithm [7,89] or forward-backward algorithm, is usually employed to estimate both the output distributions and transition probabilities. SPHINX-II uses the Baum-Welch algorithm to estimate model parameters in a maximum likelihood sense. In SPHINX-II, a two-stage training approach is taken for acoustic training. The goal of the first stage is to create a output distribution mapping table for SPHINX-II. The mapping table is generated using the distribution clustering procedure [42] and a set of one-codebook discrete HMMs. The senone mapping table relates a triphone to a sequence of senone labels (states).

The second stage is to estimate the final models that share their parameters based on the mapping table generated at the first stage. The final models of SPHINX-II are gender-dependent 4-codebook phonetic SC-HMMs [37] that are composed of a pre-specified number of senones. SPHINX-II uses SC-HMMs (or tied-mixture models), in which the continuous densities for modeling the VQ codewords are assumed to be Gaussian densities with diagonal covariance matrices. The second stage starts with estimating a set of 4-codebook context-independent phonetic D-HMMs. Subsequently, the process of senonic triphone training follows by using the output distributions and transitions of D-HMMs to initialize context-dependent triphone SC-HMMs for detailed acoustical modeling. SPHINX-II employs the Baum-Welch algorithm to estimate transition probabilities, output distributions, and codebook means and variances under a unified probabilistic framework.

The optimal number of senones varies from application to application. It depends on the amount of available training data and the number of triphones present in the task. For the corpora used in this thesis (which will be described in Section 3.2.), we use 7000 senones for the ARPA Wall Street Journal with 7200 training sentences, and 1800 senones for an alphanumerical database, AN4, with 1018 training utterances.

3.1.6. Recognition

For continuous speech recognition on large-vocabulary tasks, the search algorithm needs to apply all available acoustic and linguistic knowledge to maximize the recognition performance. In order to integrate the use of complicated models like long-distance language models and between-word triphone acoustical models for large-vocabulary tasks, SPHINX-II uses a multi-pass search approach [3]. This approach is designed to use the Viterbi algorithm [105] as a fast-match algorithm, and a detailed re-scoring approach to the N-best hypothesis [95] to produce the final recognition output.

SPHINX-II is designed to exploit all available acoustic and linguistic knowledge in three search phases. In phase one a Viterbi beam search is applied in a left-to-right fashion, as a forward search, to produce best-matched word hypothesis, along with information of word end times and associate scores, using the detailed between-word triphone models and a bigram language model. Although this first phase is implemented as a part of a three-phase decoder, it can be used to as an independent decoder for recognition when no complex language models are available for later natural-language parser.

In phase two, a Viterbi beam search is performed in a right-to-left fashion, as a backward search, to generate all possible word beginning times and scores using the between-word triphone models and a bigram model. In phase three, an A* search [80] is used to produce a set of N-best hypotheses for the test utterance by combining the results of phases one and phase two with a long distance language model. SPHINX-II can support trigrams and long-distance language models in the A* search. Because the work in this thesis focuses on environmental robustness and adaptation, only the first pass is activated for all experiments conducted in the following chapters to speed up experiments.

3.2. Experimental Tasks and Corpora

To evaluate the algorithms proposed in this thesis, we used two speech corpora for all experiments, the CMU AN4 corpus (an alphanumerical database) and the Wall Street Journal (WSJ) task. The AN4 corpus was used because when we began the development of work for this thesis, it had been the most commonly used database for robustness research at CMU [1, 99, 63, 81]. In 1992, ARPA (the Advanced Research Projects Agency) began a dictation project using material from the Wall Street Journal as the official common-evaluation large-vocabulary speech recognition task. A portion of the Wall Street Journal task (WSJ) includes speech recorded in different acoustical environments, which has been used to benchmark results in environmental robustness.

3.2.1. The AN4 Database

The AN4 database [1, 99] was collected at CMU to train and evaluate speaker-independent speech recognition systems for various acoustical conditions. Because a major goal of the AN4 database had been to provide a corpus to study the effects of changes of acoustical environments, the AN4 database consists of simultaneous recording of speech samples using two different microphones, the ARPA standard close-talking Sennheiser HMD-414 microphone, and the omnidirectional desktop Crown PZM-6FS microphone. The database is named the AN4 task because the contents in this task are alphanumerical. To nurture the development of speaker-independent systems, the AN4 database was collected using a large number of speakers selected from staff and students at CMU.

Lexicon and Grammar:

The AN4 database contains strings of letters, numbers, and a few control words that are common in the context of a census task. The speakers were asked to record utterances regarding their personal information and some random letter and digit strings.

- S-M-I-T-H
- P-I-T-T-S-B-U-R-G-H
- M-O-R-E-W-O-O-D
- X-E-D-V-SEVEN
- RUBOUT-N-A-G-K-K-THREE-THIRTY-SIX

Figure 3-4. Sample sentences in the AN4 training database

Figure 3-4 lists some example utterances in the AN4 training database. The contents of AN4 can be classified into two categories: (1) The census utterances, which consist of 9 utterances per subject that provide personal information such as names, addresses, and phone numbers. Example sentences are "**S-M-I-T-H**" and "**P-I-T-T-S-B-U-R-G-H**. (2) The alphanumerical utterances, which consist of random sequences of letters, digits, and control words. Sample utterances are "**X-**

E-D-V-SEVEN" and "**RUBOUT-N-A-G-K-K-THREE-THIRTY-SIX**".

The lexicon of the AN4 task consists of 104 vocabulary entries, of which 41 items appear fewer than 10 times. The vocabulary items are highly confusable because they consist primarily of numbers and the letters of the alphabet. There are many phonetically confusable groups, such as the E-set, Eh-set, and EY-set, as well as pairs such as *thirty/thirteen*, *fifty/fifteen*, *sixty/sixteen*, and *a/eight*. Despite the small vocabulary size, this is an intrinsically difficult task at the phonetic level.

No grammar was used in the experiments for the AN4 task. Hence this task has a perplexity [56] of 104. This perplexity, in addition to the phonetic confusibility, makes the AN4 task considerably difficult.

Acoustics of the Recording Setup:

As described before, the AN4 database consists of simultaneous recordings of speech samples in stereo using two different microphones. One recording uses the Sennheiser HMD214 close-talking microphone that has been a standard microphone in various ARPA evaluation corpora, and the other is the desk-top Crown PZM6FS microphone.

The utterances were recorded in an acoustic cubicle in a CMU speech laboratory that has a high ceiling, concrete block walls, and a carpeted floor. During the recording sessions, no attempt was made to silence other users of the room, so there is 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.

Training and Testing database:

The AN4 training corpus is composed of 1018 utterances from 74 speakers, of which 53 are male and 21 are female. There are about 14 utterances from each speaker. The ratio of male to female speakers reflects that of the general CMU population.

The testing corpus contains 140 utterances from 10 different speakers (14 sentences per speaker). The testing speakers are not present in the training set. There are 7 male speakers and 3 female speakers. Both training speech and testing speech are automatically digitized at a sampling rate of 16 kHz.

3.2.2. The Wall Street Journal-based Continuous Speech Recognition (WSJ-CSR) Corpus

As spoken language technology progresses, larger and more challenging corpora need to be created to prompt advanced research. In contrast to some pre-existing corpora such as the Resource Management (RM) and Air Travel Information System (ATIS) with medium vocabularies (< 1500 words) with language model perplexities ranging from 6 to 60, the Wall Street Journal-based (WSJ) CSR tasks are designed to provide a general-purpose English, large vocabulary, and high perplexity corpus.

The WSJ tasks consist of materials based primarily on WSJ material with WSJ text from 1987-1989 [83]. In order to support different requirements of the different research foci, the WSJ tasks are designed to accommodate the issues of variable vocabularies, variable perplexities, speaker-independence, verbalized punctuation (vp) vs. non-verbalized punctuation (nvp), speaker adaptation, microphone-independence, and changing acoustical environments.

The collection of WSJ speech data used for the CSR task is a dynamic on-going process. Therefore, in this dissertation we compared the performance of our algorithms using the pilot data collected for the ARPA WSJ task, which is referred to as WSJ0. One of the motivations for this is that we were assured that the collection of data for WSJ0 would be completed before this thesis.

Lexicon and Grammar:

The WSJ tasks are designed to be scalable and they are built to accommodate vocabularies of different sizes as well as variable perplexities. Dragon Systems, Inc. provided a set of pronunciation dictionaries with 33,000 words to cover the training, and 5,000-word and 20,000-word open and closed test conditions. 8 baseline bigram language models [83] are provided for the WSJ comparative evaluation testing. The language models are characterized along three dimensions, vocabulary size (N=5K or N=20K), “closed” or “open” vocabulary (c or o) and verbalized punctuation or non-verbalized punctuation (vp or nvp) [83].

Figure 3-4 shows some sample utterances from the WSJ0 corpus.

- THE RATE ON SIX MONTH BILLS FELL TO SIX POINT SEVEN THREE PERCENT FROM SIX POINT EIGHT THREE PERCENT
- THE SALE OF THE HOTELS IS PART OF HOLIDAY'S STRATEGY TO SELL OFF ASSETS AND CONCENTRATE ON PROPERTY MANAGEMENT
- HALLMARK HOWEVER HAS SAID IT WOULD CONTINUE THESE STATIONS IN THE SPANISH LANGUAGE FORMAT
- TATE AND LYLE PAID AN AVERAGE OF ABOUT TWO HUNDRED SIXTY PENCE A SHARE FOR ITS STAKE

Figure 3-5. Sample sentences from the WSJ0 corpus.

In order to be able to run many experiments in a limited time, we evaluated our algorithms using the 5K (5000-word) closed vocabulary, non-verbalized punctuation task (referred to as 5c-nvp hereafter) in this thesis. The associated perplexity [56] of the 5c-nvp condition is 118 [83, 42].

Acoustics of Recording Setup:

The data of the WSJ0 corpus were collected from three recording sites, MIT, SRI and TI. All utterances are recorded simultaneously using two microphones. The "primary" microphone is always a member of the Sennheiser close-talking, headset-mounted, and noise-cancelling family (HMD-410 or HMD-414). For the other channel, a few alternate microphones are rotated during different recording sessions.

Table 3-1 summarizes some of the characteristics of the 16 microphones (1 primary and 15 secondary) used to collect the WSJ0 corpus at various sites. Three of the secondary microphones were recorded at MIT, two secondary microphones were recorded at TI, and twelve secondary microphones were recorded at SRI.

As the data were collected at three different sites using different secondary microphones, we expect that the WSJ corpus supports a variety of acoustical characteristics for different environments in both training and testing condition. The environmental conditions vary from site to site. For example, the MIT data were collected in an office environment, where the ambient noise level is approximately 50 dB on the A scale of a sound-level meter, while one recording room at SRI

Microphone	description
AKG D541	stand-mounted
AT&T 5400	cordless telephone handset
AT&T 720	telephone speaker phone
Crown PCC-160	cardioid, desktop
Crown PZM-6FS	desk-mounted
Nakamichi CM100	cardioid, condenser, stand
Panasonic KXT2365	telephone speaker phone
RadioShack Omni	omnidirectional, dynamic, w/ windscreen, stand
RadioShack Highball	unidirectional, dynamic, stand
RadioShack 33-1063 Tie-Pin	omnidirectional, electret, lapel
RadioShack 33-1052 tie-clip	omnidirectional, electret, lapel
Shure SM91	unidirectional, condenser, desktop
Sony ECM155	omnidirectional, condenser, lapel
Sony ECM-50PS	electret condenser, lapel
Sony ECM-55	lapel-mounted
Sennheiser HMD-410(414)	close-talking noise-cancelling, headset

Table 3-1. The secondary microphones used in the WSJ pilot corpus.

was a large, carpeted office containing bookshelves and supply cabinets with noise level of 46 to 48 dB.

Training and Testing database:

In our studies, we use the official speaker-independent training corpus, referred to as “WSJ0-si_trn”, supplied by the National Institute of Standards and Technology (NIST) containing 7240 utterances of read WSJ text. These sentences are recorded simultaneously using two microphones. To train the recognition system, we only employ the data collected from the primary microphone, *i.e.*, a Sennheiser close-talking noise-cancelling headset. The training utterances are collected from 84 speakers at MIT, SRI, and TI. After eliminating inappropriately recorded data, we use 3651 utterances from female speakers and 3534 utterances from male speakers to train the SPHINX-II system.

To evaluate our compensation algorithms for environmental variability, we use the secondary-microphone data from various environments using the evaluation set of November 1992, referred to as “WSJ0-si_evl5”, and the SPHINX-II system trained on the training corpus, WSJ0-si_trn. The task of “WSJ0-si_evl5” consists of 330 utterances from 8 speakers (4 male and 4 female) using 3 different secondary microphones including Shure SM91, AT&T 720 telephone, and RadioShack HighBall (about 41 utterances per speaker) as shown in Table 3-2. Unless specified otherwise, this evaluation set is the default testing corpus used in our CSR experiments in this dissertation.

Microphone	descriptions	# talkers	# of utts
AT&T 720	telephone speaker phone	3	125
RadioShack Highball	unidirectional, dynamic, stand	2	82
Shure SM91	unidirectional, condenser, desktop	3	123

Table 3-2. secondary microphones in the WSJ0-si_evl5 task.

Another test set, “WSJ1-si_dt_s5”, which is the development set for the 1993 WSJ1 “Spoke 5” evaluation task, is used as a supplement to the WSJ0-si_evl5 task for our evaluation. The WSJ1-si_dt_s5 task comprises 216 sentences from 10 speakers (5 male and 5 female) using 9 different microphones (about 21 utterance per speaker) as shown in Table 3-3. The data of both WSJ0-si_evl5 and WSJ1-si_dt_s5 tasks were collected at SRI. Both training speech and testing speech are digitized at a sampling rate of 16 kHz.

3.3. Statistical Significance of Differences in Recognition Accuracy

The algorithms we propose in this dissertation are evaluated in terms of recognition accuracy observed using a common standardized corpus of speech material for testing and training. Recognition accuracy is obtained by comparing the word-string output produced from the recognizer (hereafter referred to as the hypothesis) to the word string that had been actually uttered (hereafter referred to as the reference). Based on a standard nonlinear string-matching program, word error rate is computed as the percentage of errors including insertion, deletion and substitution of words

Microphone	descriptions	# talkers	# of utts
AT&T 712	telephone handset, hand-held	1	21
AT&T 720	telephone speaker phone, desktop	1	22
Audio-Technica AT853a	cardioid, condenser, stand	2	43
RadioShack 33-992D	cardioid, dynamic, hand-held	1	22
RadioShack Pro	unidirectional, dynamic, stand	1	21
SGI	lapel-mounted	1	23
Shure WL84	unidirectional, condenser, lapel	1	22
Sony ECM-K7	super-directional, electret, desktop	1	20
Sun	built-in monitor microphone	1	22

Table 3-3. Secondary microphones in the WSJ1-si_dt_s5 task.

[52].

It is important to know whether any apparent difference in performance of the algorithms is statistically significant in order to interpret experimental results in an objective manner. Gillick and Cox [28] proposed the McNemar's test and a matched-pairs test for deciding the statistical significance of recognition results. Recognition errors are assumed to be independent in the McNemar's test or independent across different sentence segments in the matched-pairs test, respectively. Picone and Doddington [86] also advocated a phone-mediated alternative to the conventional alignment of reference and hypothesis word strings for the purpose of analyzing word errors. NIST [52] has implemented several automated benchmark scoring programs to evaluate statistical significance of performance differences between systems.

Many results produced by different algorithms do not differ from each other by a very substantial margin, and it is to our interest to know whether these performance differences are statistically significant. A straightforward solution is to apply the NIST "standard" benchmark scoring program [52] to compare a pair of results. Table B-1 in Appendix B summarizes the comparisons of our major results in terms of statistical significance for the ARPA WSJ0-si_evl5 task.

In general, the statistical significance of a particular performance improvement is closely related to the differences in error rates, and it also varies with the number of testing utterances, the task vocabulary size, the positions of errors, the grammar, and the range of overall accuracy. Nevertheless, for the ARPA WSJ0-si_evl5 task with the SPHINX-II system, a rule of thumb we have observed is that performance improvement is usually considered to be significant if the absolute difference in accuracy between two results is greater than 1%. There is usually no statistically significant difference if differences in error rate are less than 0.7%.

3.4. Summary

In this chapter, we reviewed the overall structure of SPHINX-II that will be used as the primary recognition system in our study. We also summarized the major differences between SPHINX-II and SPHINX as some of our earlier work was developed using SPHINX.

We also described two different speech corpora that we employ to evaluate the performance of our algorithms in the following chapters. The first one is the CMU AN4 corpus that had been a commonly-used database at CMU for many years. The second task is the recently-collected ARPA WSJ task designed to be a common database in ARPA speech community for development and evaluation of speech technology. “WSJ0-si_trn” will be the training corpus for the WSJ tasks in our study. For evaluation, “WSJ0-si_evl5” will be the primary evaluation corpus throughout this dissertation. The second test set, “WSJ1-si_di_s5”, will be used as a supplement to WSJ0-si_evl5, similar to the CMU AN4 corpus in some work.

Chapter 4
Blind SNR-Dependent Cepstral Normalization (BSDCN)

In the previous chapter we have identified unknown linear filtering and additive noise as two major sources of degradation in speech recognition when mismatches occur between training and testing conditions. We also reviewed some of the approaches that had been pursued to cope with these problems. The ultimate goal of these algorithms is to be able to adapt the speech recognition system to new environments with high recognition accuracy, with low computational complexity, and fast adaptation.

We have found that certain acoustical pre-processing algorithms that apply environmental compensations in the form of additive corrections in the cepstral domain are particularly well suited for many of our current robustness problems. With these procedures, compensation vectors are estimated for additive noise and for the effects of linear filtering by minimizing the differences between speech from the training and testing environments. An example of these procedures are the several algorithms developed by Acero[1] to compensate for environmental mismatches based on additive corrections. Among these algorithms, the SNR-Dependent Cepstral Normalization (SDCN) algorithm was a simple approach which could achieve moderate recognition accuracy. Unfortunately, SDCN required the use of a training database of simultaneously-recorded speech samples in the training and testing environments, so this algorithm could not adapt to unknown environments.

In this chapter we describe blind SDCN (BSDCN), an approach developed with the intent of extending the approach taken by SDCN so that environmental compensation can be applied to unknown acoustical environments. We note that the approach of BSDCN was first conceived and implemented by Acero in 1991. Additional experiments were then carried out to evaluate recognition performance on various tasks, to investigate other extensions for further improvements, and to examine the issue of amount of data for dynamic adaptation by Liu *et al* [63]. These modifications include automatic determination of ranges of instantaneous frame SNRs for different environments, refinement of the nonlinear warping function, and applications of smoothing functions to the normalization process as well as to the estimation process.

4.1. A Degradation Model for Cepstral Normalization

As in previous work on environmental compensation [1, 2], it is assumed that the observed noisy signal $y[m]$ can be modeled as a signal $x[m]$ passing through an unknown linear filter $h[m]$ whose output is then corrupted by uncorrelated additive noise $n[m]$ as shown in Figure 4-1.

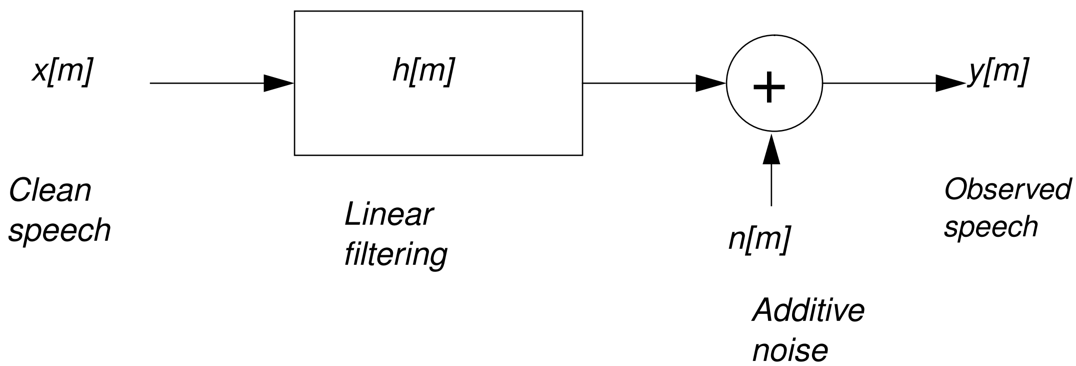


Figure 4-1. Model of signal degradation by linear filtering and additive noise.

We characterize the power spectral density (PSD) of the processes as

$$P_y(\omega) = P_x(\omega)|H(\omega)|^2 + P_n(\omega) \tag{4.1}$$

If we let the cepstral vectors $\boldsymbol{x}$, $\boldsymbol{n}$, $\boldsymbol{y}$ and $\boldsymbol{q}$ represent the Fourier series expansions of $\ln P_x(\omega)$, $\ln P_n(\omega)$, $\ln P_y(\omega)$ and $\ln|H(\omega)|^2$ respectively, Equation (4.1) can be rewritten as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{q} + \boldsymbol{r}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{q}) \tag{4.2}$$

where the correction vector $\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{q})$ is given by

$$\boldsymbol{r}(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{q}) = IDFT\{\ln(1 + e^{DFT[\boldsymbol{n} - \boldsymbol{q} - \boldsymbol{x}]})\} \tag{4.3}$$

We can obtain an estimate $\hat{P}_y(\omega)$ of the PSD $P_y(\omega)$ from a sample function of the process $y[m]$ for a frame of degraded speech that is assumed to be locally stationary. If $\boldsymbol{z}$ represents the Fourier expansion of $\ln \hat{P}_y(\omega)$, our goal is to estimate the uncorrupted vectors $\boldsymbol{X} = \boldsymbol{x}_0, \ldots\ldots, \boldsymbol{x}_{N-1}$ of an utterance given the observations $\boldsymbol{Z} = z_0, \ldots\ldots, z_{N-1}$.

4.2. Review of SDCN

The SNR-Dependent Cepstral Normalization (SDCN) algorithm [1] assumes that the correction vector depends only on the instantaneous SNR of the signal, $\boldsymbol{z}_t[0]$-$\boldsymbol{n}[0]$, so that it applies an average correction to all spectral shapes with the same SNR. An estimate, $\hat{\boldsymbol{x}}_t$, for the uncorrupted signal is obtained using the expression

$$\hat{\boldsymbol{x}}_t = \boldsymbol{z}_t - \boldsymbol{w}(SNR) . \tag{4.4}$$

These compensation vectors $\boldsymbol{w}(SNR)$ were estimated by computing the average difference in cepstra between simultaneous "stereo" recording of speech samples from the training and testing environments at each SNR of speech in the testing environment. That is, they must be "calibrated" by collecting long-term statistics from a database containing simultaneously-recorded speech samples.

At high SNRs, the correction vector primarily compensates for differences in spectral tilt between the training and testing environments (in a manner similar to the blind deconvolution procedure first proposed by Stockham et al. [100]), while at low SNRs the vector provides a form of noise subtraction (in a manner similar to the spectral subtraction algorithm first proposed by Boll [10]).

The SDCN algorithm is simple and efficient, but for every new acoustical environment encountered it must be calibrated by collecting long-term statistics from a database containing these simultaneously-recorded speech samples. In many situations such a database is impractical or unobtainable, and SDCN might not able to provide sufficiently detailed characterization of environmental variability since only long-term averages are used.

Compared to the CDCN algorithm [1], the SDCN algorithm derives its compensation vectors entirely from empirical observations of differences between data obtained from the training and testing environments. The CDCN algorithm, on the other hand, depends on a greater amount of structural knowledge about the nature of the degradations to the speech signal in order to achieve good recognition accuracy.

4.3. The Blind SDCN Algorithm

In the Blind SNR-Dependent Cepstral Normalization (BSDCN) algorithm, the need for stereophonic data is circumvented by lumping all data together at each SNR. A correspondence is established between SNRs in the training and testing environments by use of traditional nonlinear warping techniques [93] on the SNR histograms for the two environments.

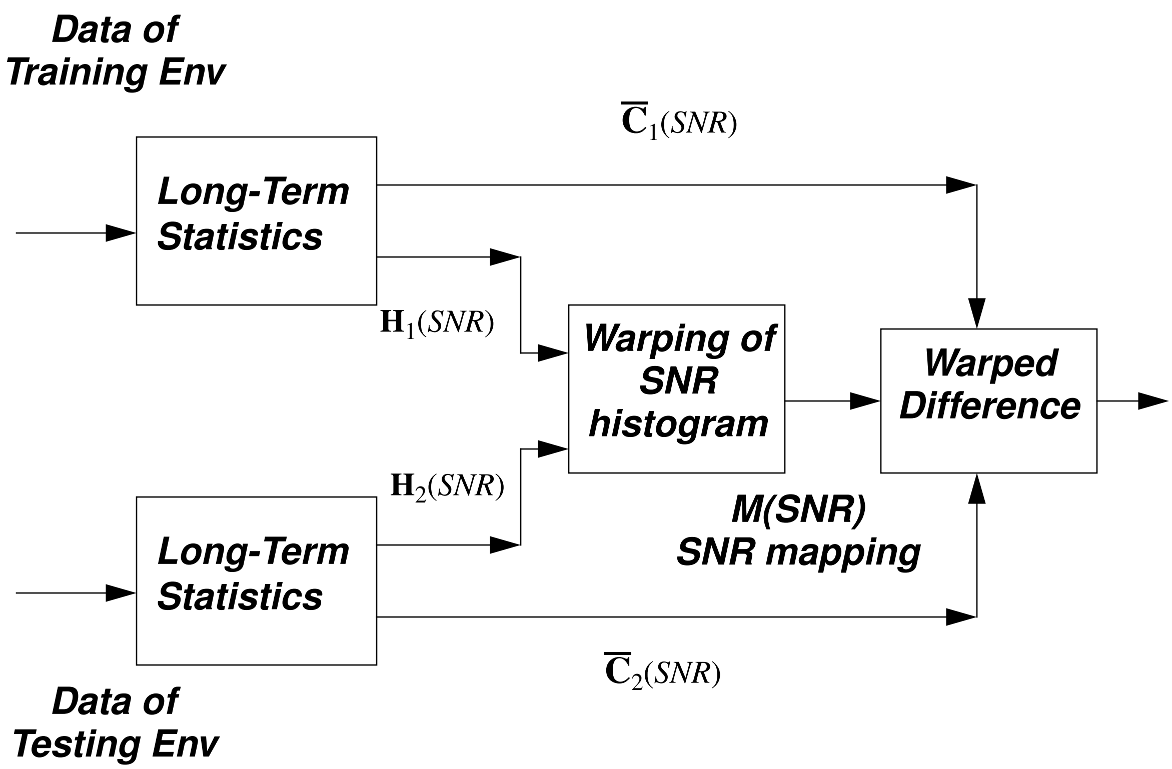


Figure 4-2. Estimation of SNR-dependent compensation vectors in BSDCN.

Figure 4-2 shows the procedure to estimate compensation vectors in BSDCN. Long-term statistics are collected for acoustical environments to produce a histogram of frame SNR values, $\mathbf{H}_1(SNR)$ and $\mathbf{H}_2(SNR)$, and a set of centroid vectors associated with each SNR value, $\bar{c}_1[SNR]$ and $\bar{c}_2[SNR]$. In general, different utterances can be used to estimate these parameters for different acoustical environments. BSDCN does not require that the data used to estimate the long-term statistics for individual environments be the same in terms of phonetic content.

Since the SNR values of the utterances in the testing environment are crucial in determining the appropriate additive vectors during compensation, a relationship between the SNR values of the training environment and those of the testing environment must be established. To achieve this goal, a nonlinear warping technique is employed to derive a mapping of SNRs, $\boldsymbol{M}(SNR)$, based on the histograms of SNR values.

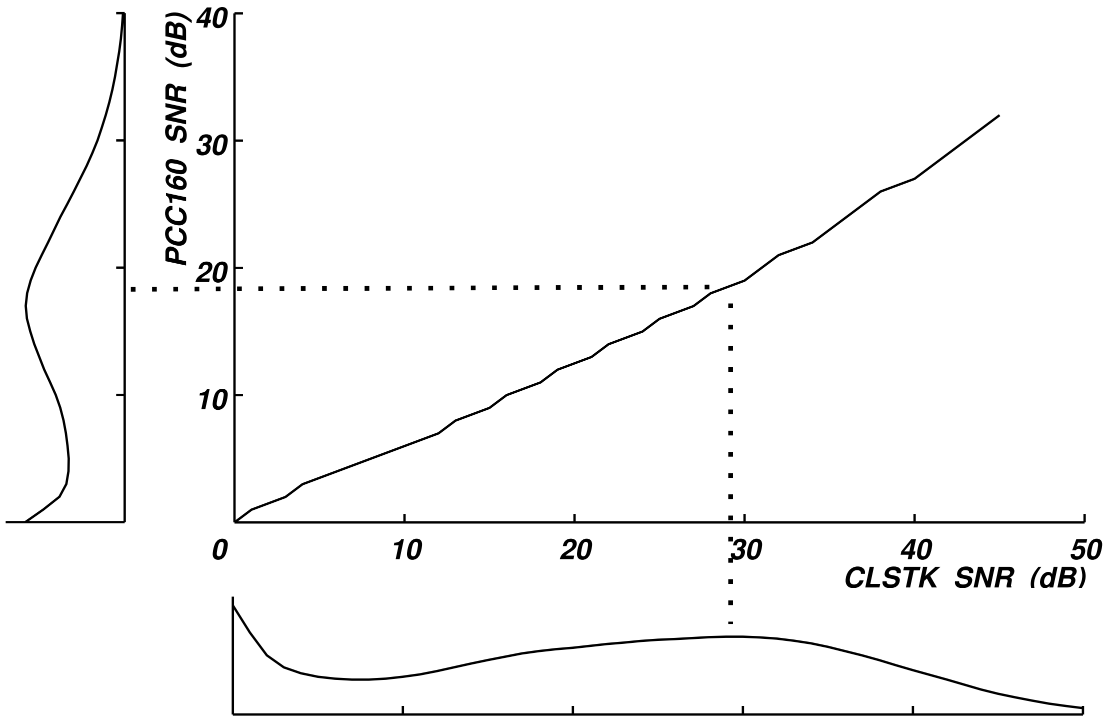


Figure 4-3. Illustration of nonlinear mapping of SNRs for the CLSTK and PCC160 microphones based on histogram of SNR values. The unlabeled graphs along the horizontal and vertical axes indicate the relative likelihood of observing various SNRs for the two microphones. The central panel indicates the warping path that best matches the two functions.

The SNR-warping procedure is illustrated in schematic form in Figure 4-3. The left and the lower panels of Figure 4-3 show typical histograms of SNRs of speech collected using a desktop-cardioid Crown PCC160 microphone (PCC160) and the close-talking Sennheiser HMD-414 (CLSTK) microphones, respectively. The central panel of Figure 4-3 shows the warping path used to match SNRs from the two microphones. As can be seen in Figure 4-3, the mode in the SNR histogram for the CLSTK microphone at 29 dB is approximately matched to the mode in the SNR histogram for the PCC160 microphone which actually occurs at 17 dB.

Calculation of the compensation vectors for BSDCN is accomplished by determining the average cepstral vector at each SNR in the training and testing environments, along with the histograms of SNRs as shown in Figure 4-3. Once a correspondence is established between the SNRs in the training and testing environments, compensation vectors are computed as the difference between average cepstra for every SNR in the testing environment and its corresponding SNR in the training environment.

There are two implementation issues that need to be noted about BSDCN. First, the histograms of SNR values must be normalized for equal area to avoid the bias of mapping the output by the environment from which more data has been collected. The minimum and maximum slopes of the warping path are currently limited to 0.2 dB/dB and 5 dB/dB, respectively, and the warping procedure seeks to minimize the Euclidean distance between the two histograms. Second, the alignment obtained by dynamically warping the SNR histograms of the training and testing data is not perfect because of the limited amount of data used to build SNR histograms and because of the slope constraints imposed in the DTW algorithm. We have found that it is beneficial to smooth the correction vectors using the simple function

$$\begin{aligned} \text{smoothed} \quad \tilde{\boldsymbol{v}}(SNR) = {} & 0.40\boldsymbol{v}(SNR) + 0.24\boldsymbol{v}(SNR+1) + 0.24\boldsymbol{v}(SNR-1) \\ & + 0.06\boldsymbol{v}(SNR+2) + 0.06\boldsymbol{v}(SNR-2) \end{aligned} \tag{4.5}$$

where $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$ refers to an arbitrary cepstral vector and its smoothed output from either environment. The weighting factors are chosen to fit in a Gaussian window, and SNR is in decibels with quantization step sizes of 1 dB.

4.3.1. Modifications for Improved Performance

The original implementation of BSDCN employed a fixed range of instantaneous frame SNRs of 30 dB with 1-dB step sizes, based on observations using the Crown PZM microphone in the AN4 database [1]. Different environments may have dramatically different dynamic ranges. For example, recordings of speech using the Sennheiser HMD414, Crown PCC160, Crown PZM, Shure SM91, and AT&T 5400 microphones in the WSJ0 training corpus exhibit ranges of frame SNRs of 41 dB, 32dB, 15 dB, 24 dB, and 32 dB, respectively. Hence, the estimation of compensation vectors can be improved for very low or very high SNRs by automatically setting the range of frame SNRs. In the ARPA WSJ0-si_evl5 task, the word error rate of BSDCN is reduced from 20.8% using a fixed range of SNRs (30 dB) to 19.3% using an automatic setting of SNR ranges in which the upper 10% and lower 10% SNRs in the SNR histogram are excluded.

A second refinement of BSDCN concerns the amount of local constraint in the non-linear warping function [93]. It was found that the original non-linear warping function could produce an unreasonable alignment path in some situations when a great deal of adjustment is required for correct alignment. The introduction of a scoring penalty rather than an equally-weighted match to the warping paths that are vertical or horizontal (when plotted as in Figure 4-3) improves the match of SNRs in these situations.

The third refinement is the application of smoothing functions in the normalization process in addition to the estimation process. We note that the improvement from the second and the third refinements was very small with only about 2% error reduction in the experiments using the AN4 task and the SPHINX system. Nevertheless, these refinements are used because they may be useful in other applications.

4.3.2. Performance of BSDCN Using SPHINX-II in CSR WSJ Tasks

Figure 4-4 illustrates results obtained using BSDCN in the ARPA CSR task using SPHINX-II. In order to cross verify the performance of BSDCN using the SPHINX-II system, we utilize the ARPA WSJ1-si_dt_s5 task as a supplement to the WSJ0-si_evl5 test set, as described in Chapter 3. Cepstral mean normalization (CMN) is employed in the SPHINX-II recognition system for each of the two experiments. Results for the WSJ0 si_evl5 task are tabulated in Table 4-1. We note from Figure 4-4 that BSDCN produces a moderate reduction of error rates in the two tasks for utterances recorded using alternate microphones.

4.3.3. Performance of BSDCN Using SPHINX and SPHINX-II in the CMU AN4 Task

Figure 4-5 illustrates word error rates obtained using BSDCN in the context of the AN4 database. These experiments were carried out with both SPHINX[1] and SPHINX-II. The results are also

1. The results of BSDCN using SPHINX and AN4 are included here because SPHINX and AN4 were the system and database used when BSDCN was developed.

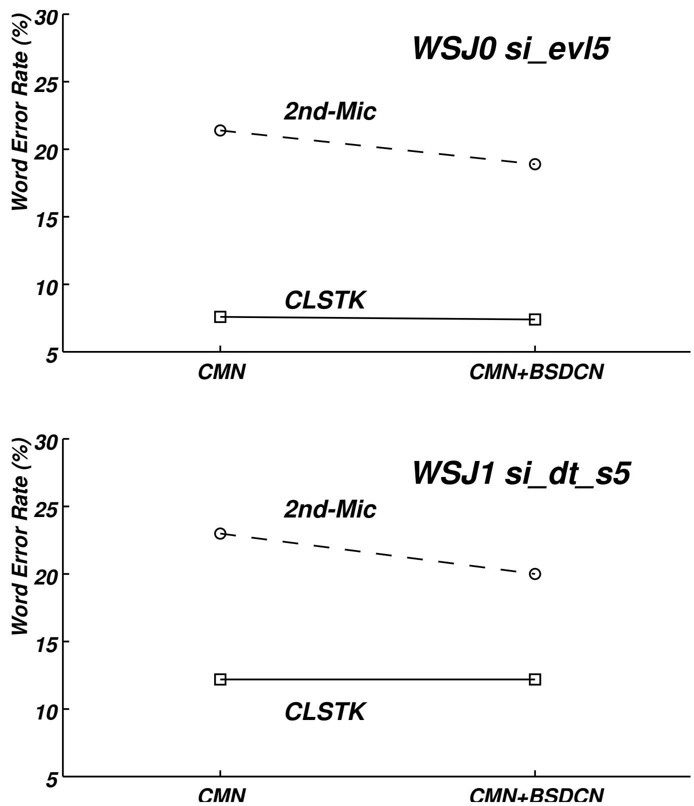


Figure 4-4. Comparisons of Blind SCDN obtained using SPHINX-II with cepstral mean normalization on the two testing corpora of ARPA CSR WSJ tasks. The upper plot is for the ARPA WSJ0-si_evl5 task and the lower is for the ARPA WSJ1-si_dt_s5 task.

Processing Algorithm	**WSJ0 si_evl5**		**WSJ1 si_dt_s5**	
	CMN	**CMN +BSDCN**	**CMN**	**CMN +BSDCN**
CLSTK (Training Mic)	7.6	7.8	12.2	12.2
Error Reduction	–	–2.6	–	0.0
Secondary Mic	21.4	19.3	23.0	20.0
Error Reduction	–	9.8	–	13.0

Table 4-1. Results of Blind SDCN using SPHINX-II with cepstral mean normalization on the testing corpus for the ARPA WSJ0- si_evl5 task.

tabulated in Table 4-2. It was found that BSDCN provided a substantial improvement by diminishing the error rate by more than 50% when no cepstral mean normalization (CMN) was used in the system for both SPHINX and SPHINX-II. Nevertheless, it can be seen that much of the improvement provided by BSDCN is also provided by the much simpler CMN algorithm, particularly in the case of the SPHINX-II recognizer. In this case, BSDCN provided a 40.0% error reduction for the SPHINX system and a 15.1% error reduction for the SPHINX-II system.

	Processing Algorithm	**No Processing**	**BSDCN**	**CMN**	**CMN +BSDCN**
SPHINX	CLSTK (Training-Mic)	13.1	13.6	12.6	13.8
	PZM6fs (Alternate-Mic)	68.6	30.0	47.7	28.6
SPHINX-II	CLSTK (Training-Mic)	14.5	14.4	13.0	13.8
	PZM6fs (Alternate-Mic)	52.3	26.0	28.5	24.2

Table 4-2. The results in word error rates for Blind SDCN on the census corpus, AN4, with two recognition systems, SPHINX and SPHINX-II. CLSTK stands for clean testing data recorded using the training microphone and PZM6fs stands for the noisy testing data recording recorded using a PZM microphone.

From the results in Table 4-1 and Table 4-2, we note the following: (1) BSDCN has low computational cost, and it can provide a moderate performance improvement for environmental mismatches. In some cases, such as the AN4 database with the SPHINX recognition system, BSDCN becomes an effective error reduction method. (2) While BSCN can achieve an error reduction of more than 40% for mismatched testing utterances in some situations, the error reduction it achieves for the SPHINX-II system with CMN is only 15.1% in the AN4 task and 10% in the WSJ task. This limited improvement is related to the fact that CMN itself is effective in compensating for some of the environmental variability using the SPHINX-II system. (3) CMN produces less improvement for SPHINX than for SPHINX-II. This may be because SPHINX-II uses the best four outputs of the VQ stage, rather than the single best output. In the AN4 task using SPHINX-II, CMN provides a 45.5% error reduction for the noisy testing data, relative to results obtained without CMN.

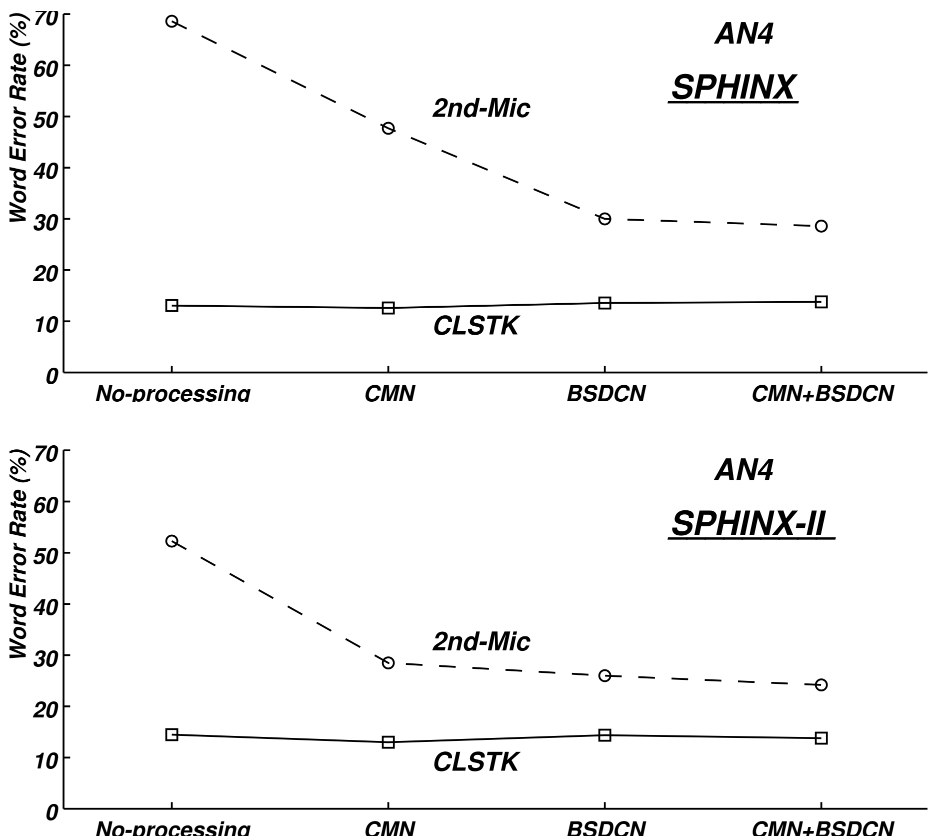


Figure 4-5. Comparison of BSDCN in the context of AN4 corpus. The upper panel illustrates the error rates obtained using SPHINX and the lower panel shows results obtained using SPHINX-II.

4.3.4. Effect of Amount of Adaptation Speech

One issue of BSDCN is how performance depends on the amount of data available for developing compensation vectors. Figure 4-6 shows how the recognition accuracy of the BSDCN algorithm depends on the amount of environment-specific speech data available for adaptation. The experiment [63] was conducted using the AN4 task and the SPHINX system, which did not employ cepstral mean normalization. The recognition system was trained on the clean speech recorded using the Sennheiser microphone and tested on the noisy data recorded using the PZM6fFS microphone. We note that the BSDCN algorithm requires about 60 seconds of adapting speech to reach asymptotic levels of recognition accuracy. This is consistent with intuition, as the BSDCN algorithm is a data-driven approach and performance can start to degrade if the amount of adaptation data becomes too small.

4.4. Summary

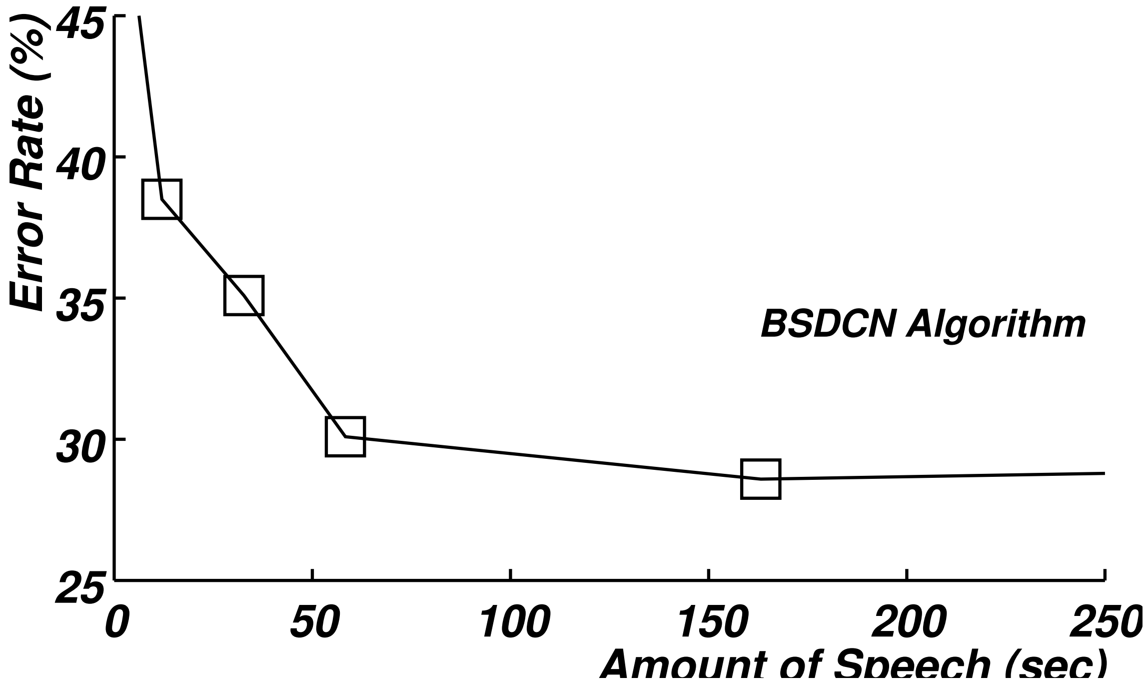


Figure 4-6. Dependence of recognition accuracy of the BSDCN algorithms on the amount of speech in the testing environment available for adaptation. The results were obtained using the AN4 task and SPHINX which was trained on Sennheiser-microphone data and tested on PZM6FS-microphone data.

In this chapter, we described the Blind SNR-dependent cepstral normalization (BSDCN) algorithm for robust speech recognition, which was originally conceived and implemented by Acero. We also described some refinements to the original BSDCN algorithm in Section 4.3.1. BSDCN compensates incoming speech for the effects of additive noise and linear filtering and it alleviates the need for stereo data in SDCN [1]. BSDCN differs from SDCN in that it does not depend on simultaneously-recorded training data to compensate for changes in environment.

When applied to the AN4 task, BSDCN produces a 50% error-rate reduction for data from the Crown PZM6FS microphone using SPHINX-II without CMN. Although CMN is helpful in compensating environmental variability, BSDCN still achieves a further reduction of 15% in errors for the AN4 task, and a 10% error reduction in the ARPA WSJ task.

Although BSDCN is able to adapt to new acoustical environments without the need for the impractical stereo recordings, its effectiveness can be limited in cases of “difficult” environments. In some environments where the noise has a high energy, the dynamic range of SNR values can become so small that the resolution of information at each SNR is reduced. Therefore, it is hard to derive a reliable SNR mapping for BSDCN. Moreover, BSDCN can also suffer in testing environments with dramatically different SNR histograms from the training environment.

In the next chapters, we will explore various other approaches that provide further increases in environmental robustness in an environment-independent fashion.

Chapter 5
Adaptation Based on
Multiple Prototype Environments

In this chapter, we propose an algorithm, Multiple Fixed Codeword Dependent Cepstral Normalization (MFCDCN), which is substantially more effective than BSDCN. MFCDCN provides a greater degree of environmental robustness based on a more detailed characterization of environmental mismatches and based on efficient use of multiple prototype environments. MFCDCN and its extension, interpolated MFCDCN (IMFCDCN), demonstrate the potential value in characterizing the variability learned from prototype environments for environmental adaptation.

5.1. Introduction

By studying simultaneous recordings from two different microphones, we have noted that a single static mapping function is usually not sufficient to characterize environmental variabilities due to changes of acoustical environments. Many approaches that have been proposed to deal with mismatched training and testing environments employ physical attributes to characterize the non-linear relationship between environments. For example, in the SNR-dependent cepstral normalization (SDCN) algorithm, the instantaneous values of frame SNR are used to distinguish different feature transformations. Similarly, compensation for environmental variabilities differs for each codeword in the codeword-dependent cepstral normalization (CDCN) algorithm.

The fixed codeword dependent cepstral normalization (FCDCN) algorithm proposed by Acero [1] characterizes environmental variability along two dimensions: the instantaneous frame SNR and VQ index. On one hand, FCDCN is very effective and efficient as it uses additive corrections to compensate for environmental mismatches. On the other hand, FCDCN is an environment-dependent algorithm that needs *a priori* knowledge to re-calibrate for every new testing environment. Specifically, FCDCN requires a set of simultaneous recording of stereo data from the particular testing environment to estimate correction vectors.

In this chapter, we propose the Multiple FCDCN (MFCDCN) and Interpolated MFCDCN (IMFCDCN) algorithms to alleviate the constraints of re-calibration of FCDCN while maintaining the effectiveness of FCDCN in compensating for acoustical environments. We begin the next section with a review of FCDCN.

5.2. Review of FCDCN

In order to characterize environmental variability, the SDCN algorithm uses frame-by-frame SNR values and CDCN employs VQ indices to partition the space of possible corrections to be applied. In terms of characterization of variability, Fixed Codeword-Dependent Cepstral Normalization (FCDCN) [2, 1] can be regarded as a combination of both SDCN and CDCN in that FCDCN makes use of both the value of the SNR and the VQ index. In terms of efficiency and effectiveness, the FCDCN algorithm provides a form of compensation that yields greater recognition accuracy than SDCN and CDCN but in a more computationally-efficient fashion than the CDCN algorithm.

5.2.1. Compensation using FCDCN

The FCDCN algorithm applies an additive correction that depends on the instantaneous SNR of the input (like SDCN), but that can also vary from codeword to codeword (like CDCN).

$$\hat{\boldsymbol{x}} = \boldsymbol{z} + \boldsymbol{r}[k, l] \tag{5.1}$$

For each frame, $\hat{\boldsymbol{x}}$ represents the estimated cepstral vector of the compensated speech, $\boldsymbol{z}$ is the cepstral vector of the incoming speech in the target environment, k is an index identifying the VQ codeword, l is an index identifying the SNR, and $\boldsymbol{r}[k, l]$ is the correction vector.

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

$$\|\boldsymbol{z} + \boldsymbol{r}[k, l] - \boldsymbol{c}[k]\|^2 \tag{5.2}$$

where $\boldsymbol{c}[k]$ is the k$^{\text{th}}$ codeword of the codebook trained using speech from the training database. The new correction vectors are estimated with an EM (estimation-maximization) algorithm [72] that maximizes the likelihood of the data.

5.2.2. Estimation of FCDCN compensation vectors

In FCDCN, the probability density function of $\boldsymbol{x}$, cepstra of speech in the training environment, is assumed to be a mixture of Gaussian densities [2, 1].

$$p(\boldsymbol{x}) = \sum_{k=0}^{K-1} P[k] N(\boldsymbol{x}; \boldsymbol{c}[k], \Sigma_k) \tag{5.3}$$

where K is the size of VQ codebook.

The cepstra of the corrupted speech are modeled as Gaussian random vectors, whose variance depends also on the instantaneous SNR, l, of the input.

$$p(z|k, \boldsymbol{r}, l) = \frac{C'}{\sigma[l]} \exp\left(-\frac{1}{2\sigma^2[l]} \|z + \boldsymbol{r}[k, l] - \boldsymbol{c}[k]\|^2\right) \tag{5.4}$$

Figure 5-1 describes the training procedure of FCDCN using the EM algorithm [72]. In practice, the EM algorithm reaches convergence after 2 or 3 iterations if we choose the vectors specified by the SDCN algorithm as initial values of the correction vectors for FCDCN. We later discovered that the need for SDCN correction vectors can be eliminated by using null vectors as initial vectors. This shortcut produces the same level of recognition accuracy at the expense of one or two additional iterations for convergence.

The computational complexity of the FCDCN algorithm is very low because changes of acoustical environment are compensated by applying additive correction vectors to the incoming testing data. In previous studies [2,63] it was found that the FCDCN algorithm provided a level of recognition accuracy that exceeded what was obtained with all other algorithms, including CDCN. However, calculation of the compensation vectors in FCDCN does require simultaneously-recorded data from the training and testing environments. In the next sections, we propose algorithms developed to alleviate the constraint of a priori knowledge of acoustical environments and requirement of stereo data for the specific testing environments before the recognition process.

5.3. Multiple Fixed Codeword-Dependent Cepstral Normalization (MFCDCN)

Multiple fixed codeword-dependent cepstral normalization (MFCDCN) is a simple extension to the FCDCN algorithm, with the goal of exploiting the simplicity and effectiveness of FCDCN

1. Assume initial values for $\boldsymbol{r}[k, l]$ and $\sigma^2[l]$.

2. **Estimate** $f_t[k]$, the *a posteriori* probabilities of the mixture components given the correction vectors $\boldsymbol{r}[k, l_t]$, variances $\sigma^2[l_t]$, and codebook vectors $\boldsymbol{c}[k]$

$$f_t[k] = \frac{\exp\left(-\frac{1}{2\sigma^2[l_t]}\|\boldsymbol{z}_t + \boldsymbol{r}[k, l_t] - \boldsymbol{c}[k]\|^2\right)}{\sum_{p=0}^{K-1}\exp\left(-\frac{1}{2\sigma^2[l_t]}\|\boldsymbol{z}_t + \boldsymbol{r}[p, l_t] - \boldsymbol{c}[p]\|^2\right)}$$

where l_t is the instantaneous SNR of the t^{th} frame.

3. **Maximize** the likelihood of the complete data by obtaining new estimates for the correction vectors $\boldsymbol{r}[k, l]$ and corresponding $\sigma[l]$:

$$\boldsymbol{r}[k, l] = \frac{\sum_{t=0}^{N-1}(\boldsymbol{x}_t - \boldsymbol{z}_t) f_t[k]\,\delta[l - l_t]}{\sum_{t=0}^{N-1} f_t[k]\,\delta[l - l_t]}$$

$$\sigma^2[l] = \frac{\sum_{t=0}^{N-1}\sum_{k=0}^{K-1}\|\boldsymbol{x}_t - \boldsymbol{z}_t - \boldsymbol{r}[k, l]\|^2 f_t[k]\,\delta[l - l_t]}{\sum_{t=0}^{N-1}\sum_{k=0}^{K-1} f_t[k]\,\delta[l - l_t]}$$

4. Stop if convergence has been reached, otherwise go to Step 2.

Figure 5-1. The training algorithm of FCDCN.

but without the need for environment-specific training. In MFCDCN, compensation vectors are precomputed in parallel for a set of target environments, using the FCDCN procedure as described above. When an utterance from an unknown environment is input to the recognition system, compensation vectors computed using each of the possible target environments are applied successively, and compensation vectors from the most likely environment based on some figure of merit are applied to the incoming utterance.

The success of MFCDCN depends on the availability of training data with stereo pairs of speech recorded from the training environment and from a variety of possible target environments, and on the extent to which the environments in the training corpus are representative of what is actually encountered in testing. With the rich acoustical characteristics for different acoustical environments, the ARPA WSJ0-si_trn training corpus presents a good database to develop and evaluate environment compensation algorithms such as MFCDCN. In our work we use each distinct environment in the ARPA WSJ0 corpus as prototype testing environments used to train compensation vectors. Since FCDCN compensation vectors are developed for each prototype testing environment, these algorithms can be viewed as extensions of FCDCN but without the need for recalibration.

5.3.1. Characterization of Environmental Variability

Figure 5-2 illustrates some typical compensation vectors obtained when the FCDCN algorithm provides compensation for a system trained using the standard close-talking Sennheiser HMD-414 microphone, and tested using the unidirectional desktop PCC-160 microphone as the target environment. The vectors are computed for 8 VQ clusters at the extreme SNRs of 0 and 29 dB, as well as at 5 dB. These curves are obtained by calculating the cosine transform of the cepstral compensation vectors, so they provide an estimate of the effective spectral profile of the compensation vectors. The horizontal axis represents frequency, warped nonlinearly according to the mel scale [17]. The maximum frequency corresponds to the Nyquist frequency, 8000 Hz.

We note that the spectral profile of the compensation vector varies with SNR, and that the various VQ clusters require compensation vectors of different spectral shapes, especially for the intermediate SNRs from 5 to 10 dB. Each VQ cluster produces a different compensation curve for each SNR.

Similarly, Figure 5-3 illustrates typical compensation vectors obtained when the FCDCN algorithm provides compensation for a system trained using the standard close-talking Sennheiser HMD-414 microphone, and tested using a telephone speakerphone, the AT&T 720 microphone, as the target environment.

Both Figure 5-2 and Figure 5-3 demonstrate that instantaneous frame SNRs and VQ indices help discriminate the environmental variability with the same target testing environments. An interpretation for these two features is that the instantaneous frame SNR helps distinguish global environmental difference between various signals while the VQ index helps to refine the distinction of speech within each SNR cluster. A comparison between Figure 5-2 and Figure 5-3 reveals fundamental differences in acoustical characteristics between the two distinct target environments.

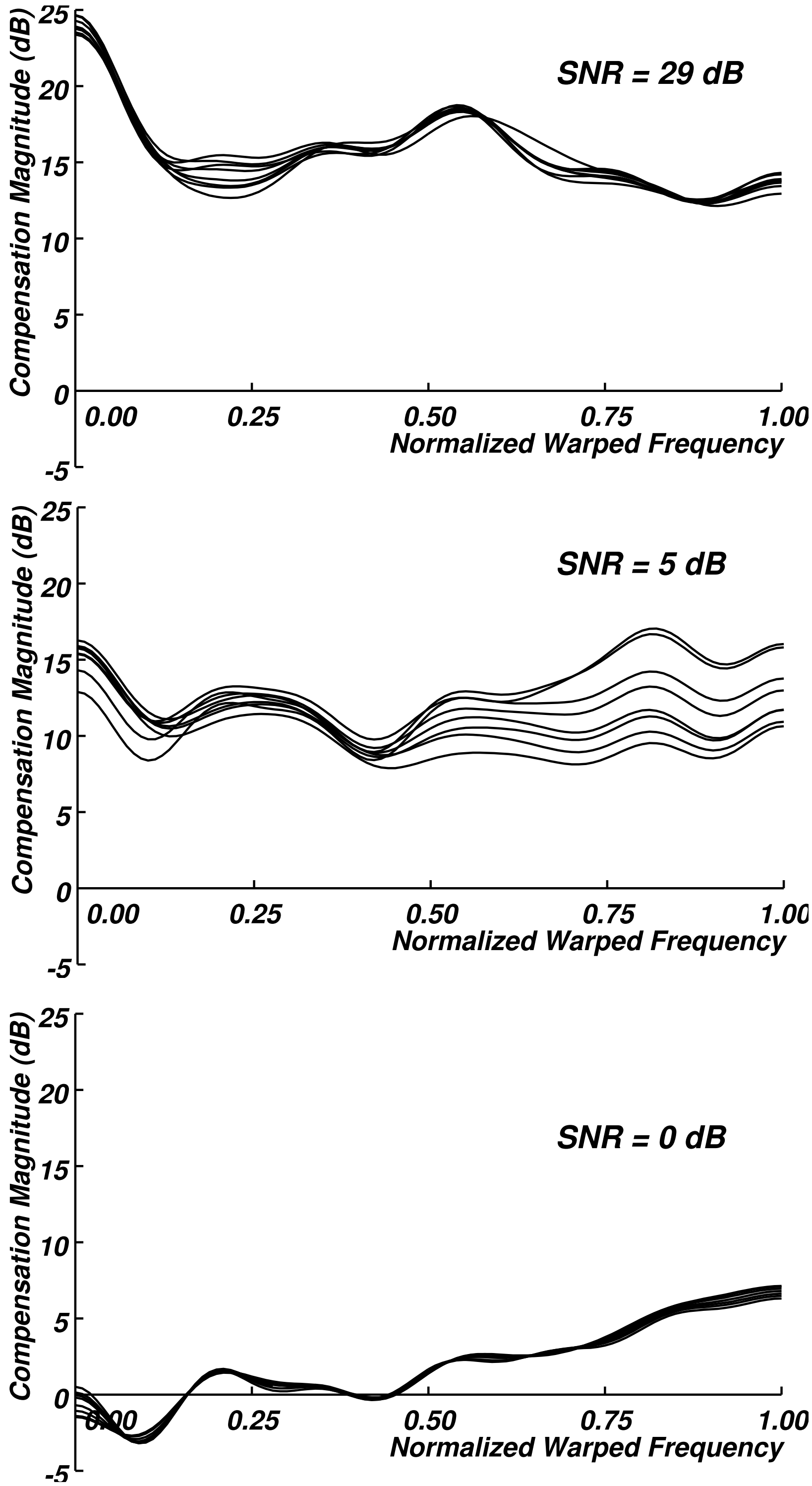


Figure 5-2. Comparison of compensation vectors using the FCDCN method with the PCC-160 unidirectional desktop microphone, at three different signal-to-noise ratios. The maximum SNR used by the FCDCN algorithm is 29 dB.

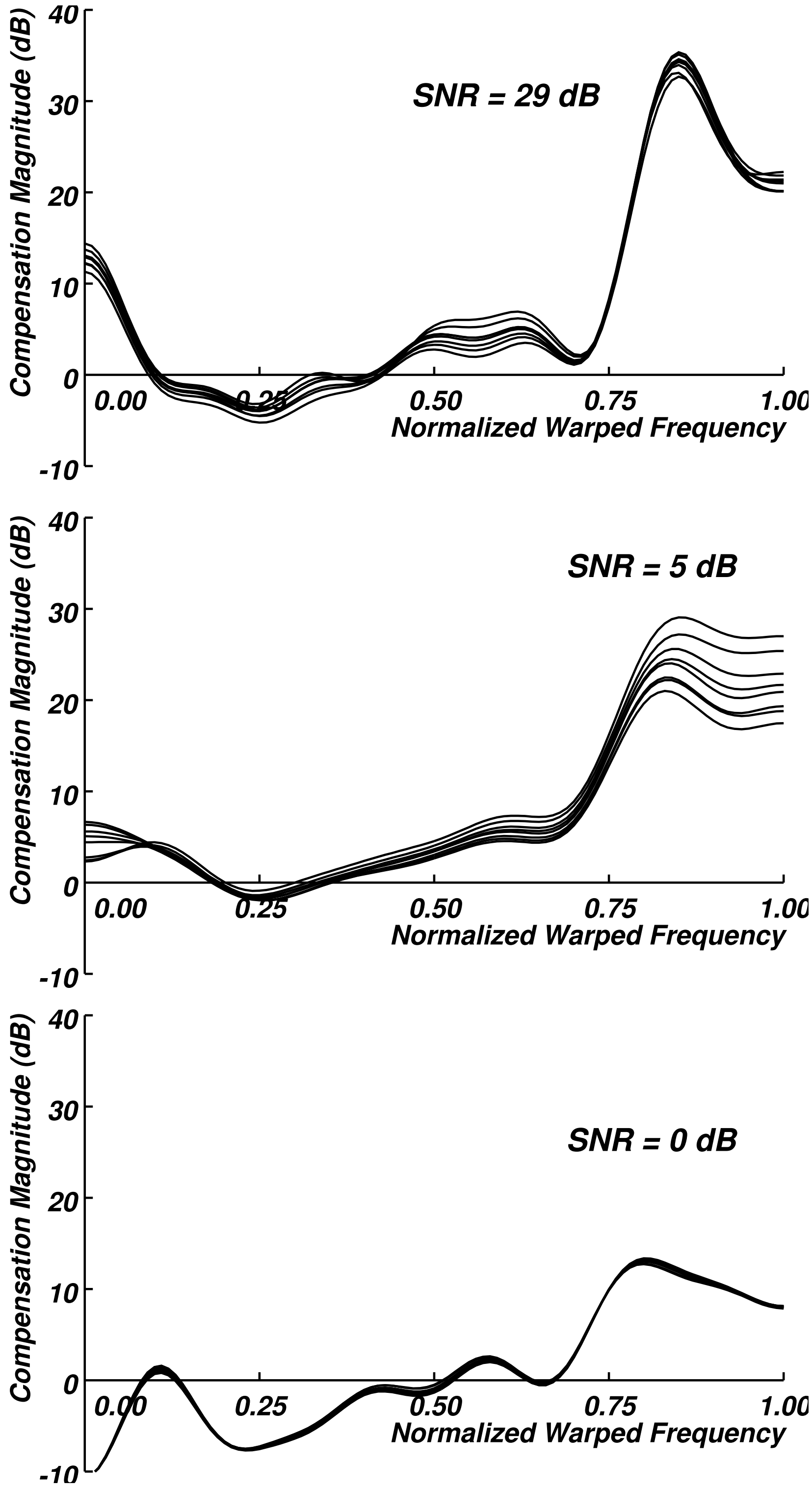


Figure 5-3. Comparison of compensation vectors using the FCDCN method with the AT&T 720 speakerphone, at three different signal-to-noise ratios. The maximum SNR used by the FCDCN algorithm is 29 dB.

These differences normally would require FCDCN to re-calibrate for each new target environment, while the use of MFCDCN avoids the need for this recalibration.

5.3.2. Estimation of Compensation Vector for MFCDCN

The goal of the training process of MFCDCN is to estimate compensation vectors for each prototype testing environment in the corpus. For each prototype environment, the estimation algorithm is the same as FCDCN as described in Section 5.2.

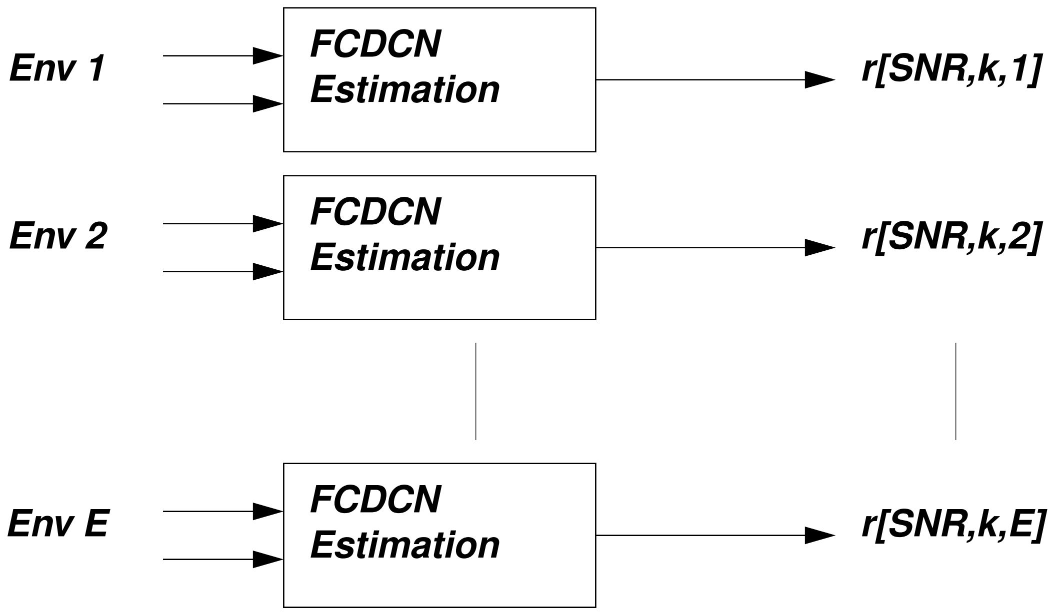


Figure 5-4. The training process for MFCDCN. Each block represents a training procedure of FCDCN for each of the prototype environments in the training corpus. *E* is the total number of acoustical environments.

Figure 5-4 is a block diagram of the training process for MFCDCN. Each block constitutes an estimation process for each prototype environment using the FCDCN algorithm. Thus, compensation vectors are precomputed in parallel for later use in the recognition phase. Each environment in the ARPA WSJ0 training corpus is used to estimate the compensation vectors based on simultaneous recordings of speech in the training environment (*i.e.* the "clean" Sennheiser HMD-414) and the prototype testing environment.

Although the MFCDCN compensation vectors are derived in supervised mode using the FCDCN training algorithm, these compensation vectors are applied during recognition in unsupervised mode using one of several microphone selection algorithms. These compensation vectors can also be employed in other unknown acoustical environments. In the current implementation of MFCD-CN, we use 16 different prototype acoustical environments from the ARPA WSJ0 training corpus, including the standard training Sennheiser close-talking microphone.

5.3.3. Environment Selection

In this section, we discuss two procedures that are used for environment selection in this dissertation, selection by compensation and the Gaussian environment classifier [66].

5.3.3.1. Selection by Compensation

In the selection-by-compensation method, compensation vectors computed using each of the prototype testing environments are applied successively to the incoming utterance, and the environment is chosen that minimizes the residual VQ distortion over the entire utterance. Specifically, a pair of outputs from the compensation process corresponding to each prototype environment is generated. The pair consists of a residual VQ distortion over the entire utterance, D_e,

$$D_e = \sum_t^T d_{t,e} \ , \qquad \text{where} \quad d_{t,e} = \underset{k}{Min} \| z_t + r[k,l,e] - c[k] \|^2 \tag{5.5}$$

and the associated compensated speech feature, $\hat{\boldsymbol{x}}_e$,

$$\hat{\boldsymbol{x}}_e = \hat{\boldsymbol{x}}_{1,e} \hat{\boldsymbol{x}}_{2,e} \hat{\boldsymbol{x}}_{3,e} \ldots\ldots \hat{\boldsymbol{x}}_{t,e} \ldots$$
$$\text{where} \quad \hat{\boldsymbol{x}}_{t,e} = \boldsymbol{z}_t + \boldsymbol{r}[k',l,e] \quad \text{and} \quad k' = \arg \underset{k}{Min} \ \| z_t + r[k,l,e] - c[k] \|^2 \tag{5.6}$$

where k refers to the VQ codeword, l to the SNR, and e to the prototype environment used to train the ensemble of compensation vectors.

The selected environment is determined so as to minimize the residual VQ distortion over the entire utterance, D_e. Note that in this approach selection between environments is made based on the score generated during the normalization process.

5.3.3.2. The Gaussian Environment Classifier

The second procedure, the Gaussian environment classifier, models each prototype environment with mixtures of Gaussian density functions. Environment selection is accomplished by choosing the environment that would produce the original (uncompensated) test data with the greatest probability. More details about the Gaussian environment classifier can be found in Section 6.3.1.. With this approach environment selection is achieved by using the original test data without any compensation, in contrast to the selection-by-compensation procedure.

5.3.3.3. Discussion of Environment Selection

Using data from the ARPA WSJ0-si_evl5 task, the selection-by-compensation method produces environment-selection errors 32.1% of the time for data recorded using "secondary" microphones (noisy testing speech) and it has a 3.0% error rate for data obtained using the close-talking Sennheiser microphone (clean testing speech). The Gaussian environment classifier produces a 10.6% misjudgment rate for data using secondary microphones and a 11.8% error rate for Sennheiser microphone data. Detailed results of environment selection can be found in the confusion matrices shown in Appendix C.

In an attempt to understand the effect of our microphone selection procedure on the recognition accuracy, an experiment was carried out using MFCDCN on the WSJ0-si_evl5 task by assuming that the correct environment identity was given to the recognition system. We did not observe any difference in recognition accuracy between error rates obtained with blind environment selection and with perfect knowledge of the correct environment identity. This indicates that the system can still benefit from compensation vectors of acoustically similar environments.

In general, we have found no substantial difference between these two environment selection procedures in terms of ultimate recognition accuracy [66]. Because of this, we chose the environment selection procedure because it provided greater computational efficiency for a particular compensation algorithm. For some algorithms where the compensation vectors can be explicitly determined before the search, we use the selection-by-compensation procedure. For other algorithms where it is more difficult to utilize compensated output for environment selection during the search, we make use of the Gaussian environment classifier. These general approaches are similar in spirit to other approaches [96, 74].

5.3.4. Dependence of Recognition Accuracy on Amount of Data

We now consider the dependence of recognition accuracy on the amount of data used to select the compensation environment from the ensemble of prototype environments. Figure 5-5 shows how the accuracy of environment selection procedures depends on the amount of testing speech

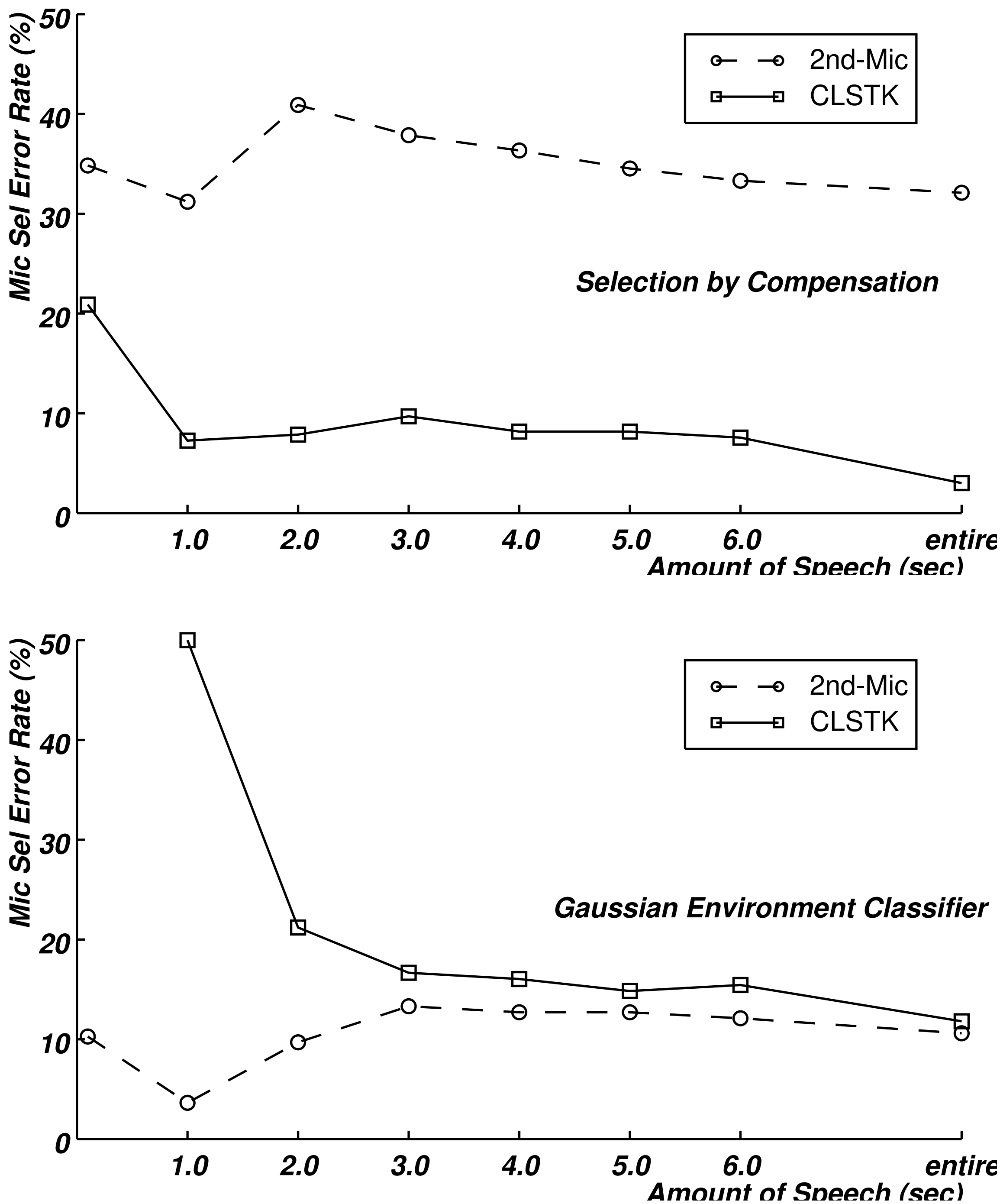


Figure 5-5. Dependence of environment selection procedures on the amount of speech used for selection. The upper and lower panels represent results from the "selection-by-compensation" and "Gaussian environment classifier", respectively.

available. The upper and lower panels compare the microphone selection results for the *selection-by-compensation* and *Gaussian environment classifier*, respectively. We observe that the accuracy of environment selection differs for clean data and noisy data. In both selection procedures, microphone selection for clean data benefits from more speech. On the other hand, microphone selection for noisy speech tends to improve when the available speech becomes less than 2 seconds. In general, environment selection results do not vary dramatically except for clean data shorter than 2 seconds.

5.3.5. Compensation using MFCDCN

As noted above, MFCDCN operates by selecting an environment among the prototype environments that is most likely to have produced the testing data. MFCDCN provides compensation for unknown environments on a sentence-by-sentence basis, not requiring that all testing utterances be recorded in the same condition. This enables MFCDCN to handle possible changes of acoustical testing conditions during the recording session.

Figure 5-6 shows the compensation procedure of MFCDCN used during recognition. Each possible prototype environment is used to match the testing utterance in turn. The best compensated output, $\hat{\boldsymbol{x}}_b$, is generated as described in Equation (5.7).

$$\hat{\boldsymbol{x}}_b = \hat{\boldsymbol{x}}_i \qquad \text{where} \quad i = \arg \underset{e}{Min} \; D_e \tag{5.7}$$

where D_e is the residual VQ distortion over the entire utterance as defined in Equation (5.5), and $\hat{\boldsymbol{x}}_i$ is the associated compensated speech feature, defined in Equation (5.6), if environment i is chosen.

Figure 5-7 compares recognition word error rates for the ARPA WSJ0-si_evl5 task. These results are obtained by applying MFCDCN to a system without using cepstral mean normalization (designated as MFCDCN) and to a baseline system that uses cepstral mean normalization (designated as CMN+MFCDCN). In the system without cepstral mean normalization, the MFCDCN produces a 56.6% reduction in word error rate compared to the baseline. For the system using cepstral mean normalization as part of the standard processing, the MFCDCN algorithm produces a 32.2%

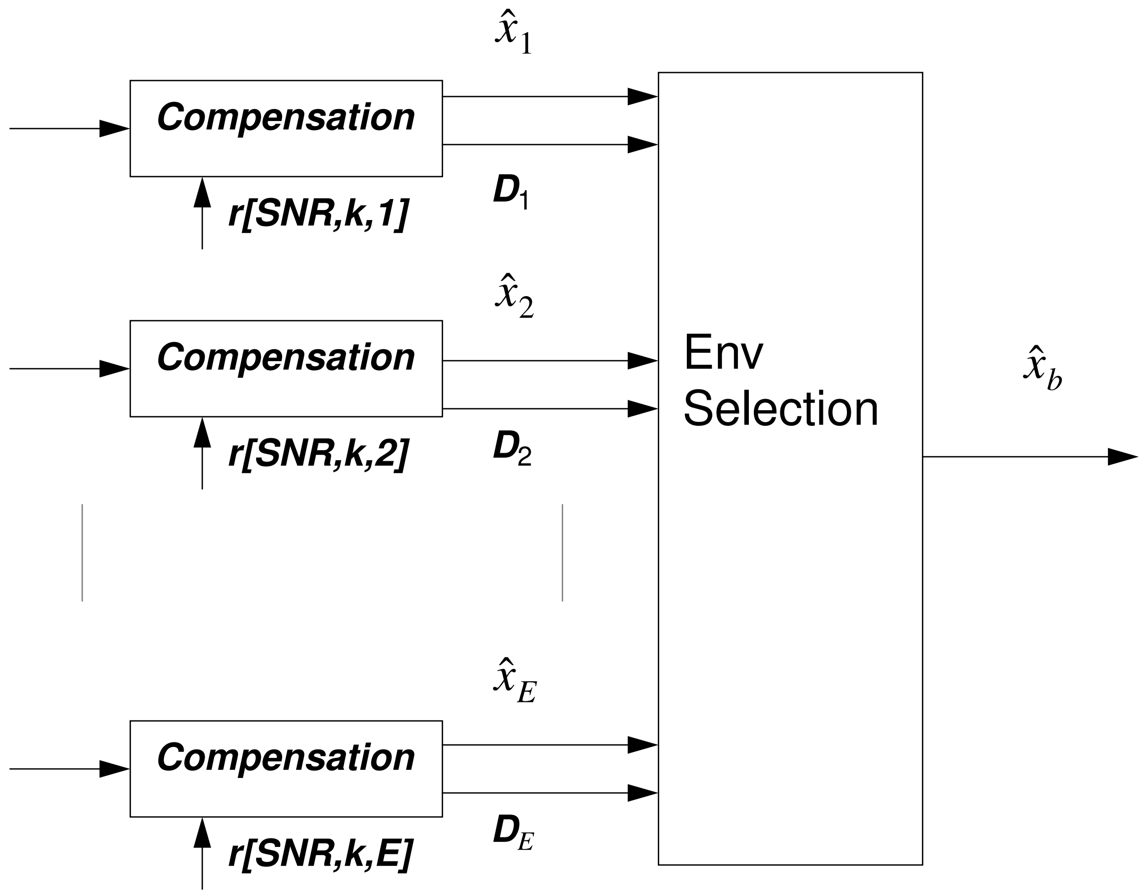
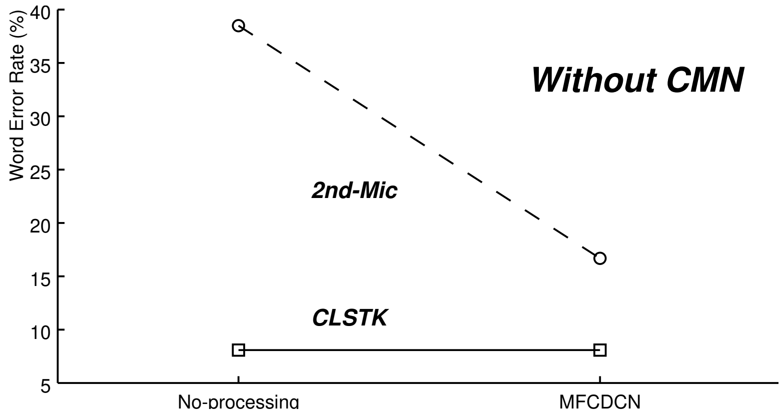


Figure 5-6. Compensation procedure for MFCDCN.

processing algorithm	No Processing	MFCDCN	CMN	CMN +MFCDCN
CLSTK (Training Mic)	8.1	8.1	7.6	7.6
Error Reduction	--	0	--	0
Secondary-Mic Data	38.5	16.7	21.4	14.5
Error Reduction	--	56.6	--	32.2

Table 5-1. Percentage of word errors and corresponding error rate reduction for MFCDCN with cepstral mean normalization on the ARPA WS0-si_evl5 task.



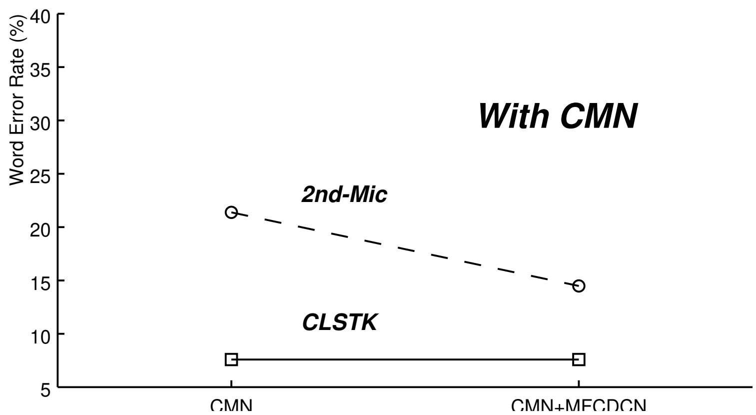


Figure 5-7. Results of MFCDCN in systems with and without cepstral mean normalization on the ARPA WS0-si_evl5 task.

error rate reduction compared to a system with cepstral mean normalization. This substantial error rate reduction shows that mismatches between the training and testing environments are alleviated dramatically, even for systems that already use cepstral mean normalization. While environment selection for the compensation vectors of MFCDCN is generally performed on an utterance-by-utterance basis, the probability of a correct selection can be further improved by allowing the classification process to make use of cepstral vectors from previous utterances in a given session as well.

The success of MFCDCN depends on the availability of training data with stereo pairs of speech recorded from the training environment and from a variety of possible target environments. It also depends on the accuracy of microphone selection and the extent to which the prototype environments in the training data are representative of what is actually encountered in testing.

We now consider the question of how performance is affected when the actual testing environment is not included in the prototype environments used to compute the compensation vectors. For this purpose, we deliberately exclude the testing microphones from the corpus used to derive the compensation vectors. In this case, cepstra of testing utterances are to be normalized using compensation vectors from the most acoustically similar but incorrect prototype environment. Table 5-2 compares results from the ARPA WSJ0-si_evl5 task when the actual testing environment is excluded from the prototype environments. These result show a 12.4% increase in word error rate, from 14.5% to 16.3%, compared to the results in Table 5-1.

Processing Algorithm	**CMN**	**CMN +MFCDCN**
CLSTK (Training Mic)	7.6	7.6
Error Reduction	–	0
Secondary-Mic Data	21.4	16.3
Error Reduction	–	23.8

Table 5-2. Percentage of word errors and corresponding error rate reduction for MFCDCN with cepstral mean normalization on the ARPA WSJ0-si_evl5 task. In this particular experiment, we exclude all three microphones from the training corpus used for derivation of compensation vectors.

5.4. Interpolated Multiple Fixed Codeword Dependent Cepstral Normalization (IMFCDCN)

The MFCDCN algorithm described above applies compensation from the single environment in the training set that is believed to have acoustical characteristics that most closely resemble those of the testing environment. In some cases, however, the testing environment does not closely resemble any single environment in the training set. We showed in the last section that accuracy degrades somewhat under these circumstances.

To alleviate the this problem, we propose an algorithm that estimates a new compensation vector for testing data on an utterance-by-utterance basis by interpolating among the compensation vectors calculated for each individual prototype environment. In cases where no acoustically similar compensation vectors exist, interpolating the compensation vectors of several environments may be more helpful than using compensation vectors from any single (incorrect) environment. We refer to this procedure as interpolated multiple fixed codeword-dependent cepstral normalization, or IMFCDCN.

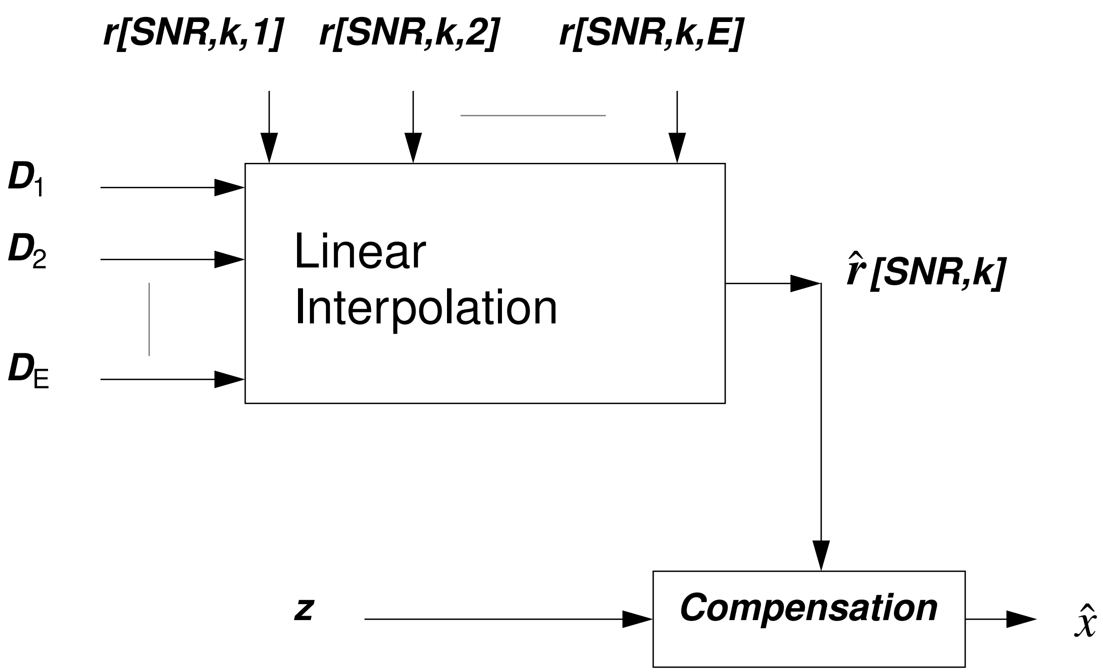


Figure 5-8. Block diagram of Interpolated MFCDCN using an ensemble of *E* prototype environments.

Figure 5-7 describes the implementation of IMFCDCN. As in the case of MFCDCN, the testing sentence is normalized with compensation vectors from each individual prototype environment using FCDCN. The resultant residual VQ distortion over the entire utterance, D_e, for environment e is defined in Equation (5.5), as in MFCDCN.

The weighting factors used for interpolation across environments in the IMFCDCN algorithm are proportional to the probability of a particular environment given the noisy speech as shown in Equation (5.8)

$$f_e \propto p(e|\bar{\mathbf{Z}}) = \frac{p(\bar{\mathbf{Z}}|e)\,p(e)}{p(\bar{\mathbf{Z}})} \propto p(\bar{\mathbf{Z}}|e) \tag{5.8}$$

where all environments are assumed to be equally probable. The weighting factors for linear interpolation can be re-written as

$$\sum_e f_e = 1 \quad \Rightarrow \qquad f_e = \frac{p(e|\bar{\boldsymbol{Z}})}{\sum_{i=1}^{E} p(i|\bar{\boldsymbol{Z}})} = \frac{p(\bar{\boldsymbol{Z}}|e)}{\sum_{i=1}^{E} p(\bar{\boldsymbol{Z}}|i)} \tag{5.9}$$

The *pdf* of the CLSTK speech given mixture k is defined as

$$p(\boldsymbol{x}_t) = \frac{C}{\sigma}\exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_t - \boldsymbol{c}[k]\|^2\right) \tag{5.10}$$

where $\boldsymbol{c}[k]$ and σ are mean vectors and the standard deviation of the codebook trained on clean speech and $\boldsymbol{x}_t$ is the cepstral vector at time t. The observed noisy speech from environment e is modeled as a Gaussian random vector [1] as shown in Equation (5.11) if the value of $\boldsymbol{x}_t$ is known,

$$p(z_t|\boldsymbol{x}_t,\ k,\ SNR{=}l_t\ , \boldsymbol{r}[k, l_t, e]) = \frac{C''}{\sigma_e[l_t]}\exp\left(\frac{-1}{2\sigma_e^2[l_t]}\|z_t + \boldsymbol{r}[k, l_t, e] - \boldsymbol{x}_t\|^2\right) \tag{5.11}$$

and as shown in Equation (5.12) if no knowledge is available on $\boldsymbol{x}_t$

$$p(z_t|k,\ SNR{=}l_t\ , \boldsymbol{r}[k, l_t, e]) = \frac{C''}{\sigma_e[l_t]}\exp\left(\frac{-1}{2\sigma_e^2[l_t]}\|z_t + \boldsymbol{r}[k, l_t, e] - \boldsymbol{c}[k]\|^2\right) \tag{5.12}$$

Based on Equation (5.12), we can define the *pdf* of the observed speech, $\boldsymbol{z}_t$, given environment e as

$$p(z_t|e) = p(z_t|k{=}k',\ SNR{=}l_t\ , \boldsymbol{r}[k, l_t, e]) = \frac{C''}{\sigma_e[l_t]}\exp\left(\frac{-1}{2\sigma_e^2[l_t]}\|z_t + \boldsymbol{r}[k', l_t, e] - \boldsymbol{c}[k']\|^2\right)$$

$$= \frac{C'}{\sigma_e[l_t]}\exp\left(\frac{-1}{2\sigma_e^2[l_t]}d_{t,e}\right) \tag{5.13}$$

where k' and $d_{t,e}$ are the best Gaussian mixture and the corresponding instantaneous VQ distortion at time frame t as defined in Equation (5.5) and Equation (5.6). Therefore, the probability of the observing utterance, $\bar{\boldsymbol{Z}} = z_1 z_2 z_3 \ldots\ldots z_t \ldots\ldots z_T$, given environment e can be expressed as

$$p(\bar{\boldsymbol{Z}}|e) = \prod_t p(z_t|e) = \prod_t \frac{C'}{\sigma_e[l_t]}\exp\left(\frac{-1}{2\sigma_e^2[l_t]}d_{t,e}\right) \tag{5.14}$$

The weight factors can be re-written as

$$f_e = \frac{p(\bar{\mathbf{Z}}|\, e)}{\sum_{i=1}^{E} p(\bar{\mathbf{Z}}|\, i)} = \frac{\prod_t \frac{1}{\sigma_e[l_t]} \exp\left(\frac{-1}{2\sigma_e^2[l_t]} d_{t,e}\right)}{\sum_{i=1}^{E} \prod_t \frac{1}{\sigma_i[l_t]} \exp\left(\frac{-1}{2\sigma_i^2[l_t]} d_{t,i}\right)} \tag{5.15}$$

Another simplified variant to Equation (5.15) is to use a single variance for all SNRs and all environments. In this case, the weight factors can be expressed as

$$f_e = \frac{exp\{-D_e/(2\sigma^2)\}}{\sum_{i=1}^{E} exp\{-D_i/(2\sigma^2)\}} \tag{5.16}$$

where σ is the codebook standard deviation using clean speech and D_i and D_e are the residual VQ distortions of the i^{th} and e^{th} environments as defined in Equation (5.5). We find that the weighting factors obtained using Equation (5.16) produce slightly better recognition results than the weighting factors obtained using Equation (5.15). Our hypothesis is that it is usually hard to estimate reliably variances using a limited amount of adaptation data and that differences of variances can outweigh those of residual VQ distortions in computing environmental probabilities.

The IMFCDCN algorithm estimates compensation vectors for new environments by linear interpolation of several of the compensation vectors as:

$$\hat{\mathbf{r}}[k,l] = \sum_{e=1}^{E} f_e \cdot \mathbf{r}[k,l,e] \tag{5.17}$$

where $\hat{\mathbf{r}}[k,l]$, $\mathbf{r}[k,l,e]$, and f_e are the estimated compensation vectors, the environment-specific compensation vector for the e^{th} environment, and the weighting factor for the e^{th} environment, respectively. In the current implementation of IMFCDCN, we set E to 3 empirically.

Figure 5-9 compares results obtained using IMFCDCN and MFCDCN when all secondary microphones used in the testing utterances are excluded from the set of compensation vectors. The ARPA WSJ0-si_evl5 task is used in combination with cepstral mean normalization. Table 5-3

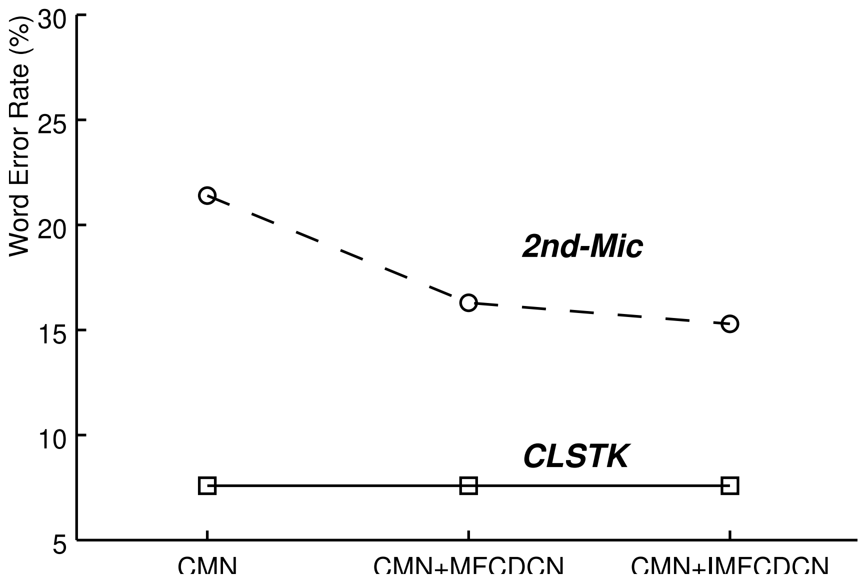


Figure 5-9. Comparison of IMFCDCN and MFCDCN in systems with cepstral mean normalization on the ARPA WSJ0-si_evl5 task. In this particular experiment, all three testing microphones are not included in the estimation process.

Processing Algorithm	**CMN**	**CMN +MFCDCN**	**CMN +IMFCDCN**
CLSTK (Training Mic)	7.6	7.6	7.6
Error Reduction	--	0	0
Secondary-Mic Data	21.4	16.3	15.6
Error Reduction	--	23.8	27.1

Table 5-3. Percentage of word errors and corresponding error rate reduction for IMFCDCN and MFCDCN with CMN on the ARPA WS0-si_evl5 task with all three testing microphones are excluded from the estimation process, corresponding to Figure 5-9.

shows that the word error rate is reduced from 16.3% using MFCDCN to 15.6% using IMFCDCN. In other words, robustness with respect to acoustical environment can be improved by interpolating compensation vectors if no compensation vectors from the actual testing environment are available. A similar reduction of error rate provided by IMFCDCN is also demonstrated in Figure 5-10 and Table 5-4 when the ARPA WSJ1-si_dt_s5 task is used as a test set.

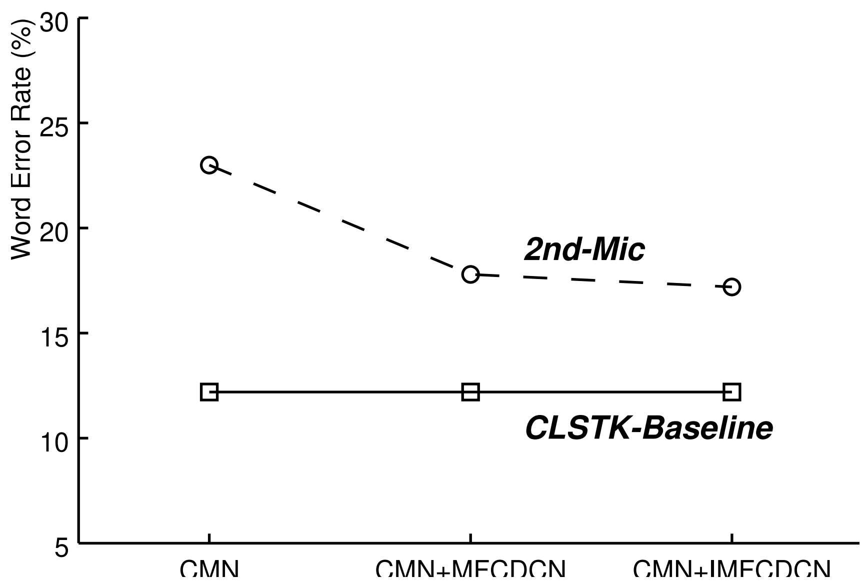


Figure 5-10. Comparison of IMFCDCN and MFCDCN on the ARPA WSJ1-si_dt_s5, in which testing microphones are not among the prototype compensation vectors.

Processing Algorithm	CMN	CMN +MFCDCN	CMN +IMFCDCN
CLSTK (Training Mic)	12.2	12.2	12.2
Error Reduction	--	0	0
Secondary-Mic Data	23.0	17.8	17.2
Error Reduction	--	22.6	25.2

Table 5-4. Percentage of word errors and corresponding error rate reduction for IMFCDCN and MFCDCN with CMN on the ARPA WS1_si_dt_s5 task, corresponding to Figure 5-10.

Using linear interpolation of compensation vectors across environments reduces the susceptibility of IMFCDCN to the effects of new unknown testing environments, compared to MFCDCN. On the other hand, we note that MFCDCN does provide better recognition accuracy in cases where the actual testing environments is one of the prototype environments, as shown in Figure 5-11 and Table 5-5. This result is not surprising if the microphone selection works properly. In a similar situation where development utterances are available for deriving utterances beforehand, linear interpolation of compensation vectors from other incorrect environments can only dilute the effect of the proper compensation vectors, which would increase error rate.

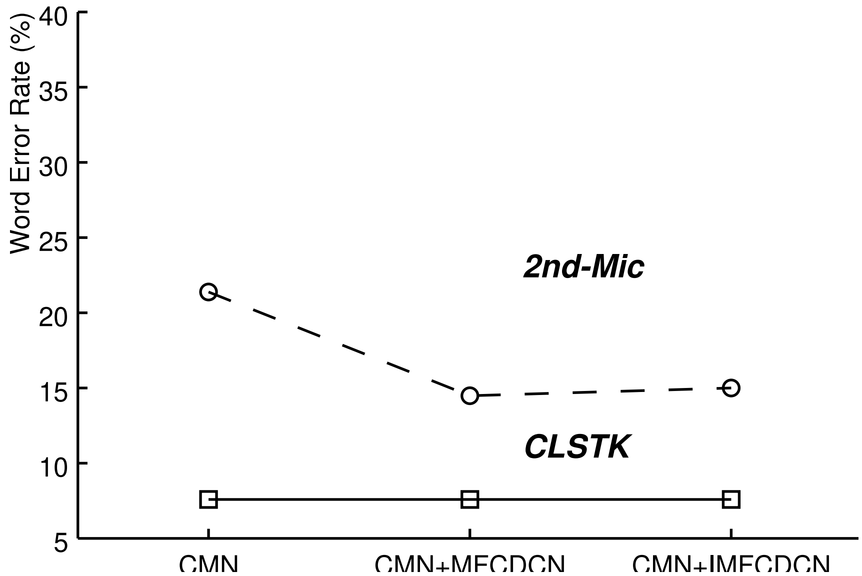


Figure 5-11. Comparison of IMFCDCN and MFCDCN on the ARPA WSJ0-si_evl5 task. This test is the same as Figure 5-9 except that we do not exclude all three microphones in this experiment.

Processing Algorithm	CMN	CMN +MFCDCN	CMN +IMFCDCN
CLSTK (Training Mic)	7.6	7.6	7.6
Error Reduction	--	0	0
Secondary-Mic Data	21.4	14.5	15.0
Error Reduction	--	32.2	29.9

Table 5-5. Percentage of word errors and corresponding error rate reduction for IMFCDCN and MFCDCN with CMN on the ARPA WS0-si_evl5 task, corresponding to Figure 5-11.

We also note that MFCDCN is a special case of IMFCDCN. If the number of prototype environments to be considered for linear interpolation is set to 1, the IMFCDCN algorithm reduces to MFCDCN.

5.5. Summary

In this chapter, we have proposed two algorithms that extend the successful environment-dependent approach of FCDCN to compensate for the effects of unknown environmental variability without the need to re-calibrate the system using stereo-data.

The Multiple Fixed Codeword Dependent Cepstral Normalization (MFCDCN) algorithm transforms testing utterances to the training acoustical space without requiring that the identity of the testing environment be known *a priori*. Using compensation vectors pre-computed in parallel from a standard suite of unknown microphones, MFCDCN is applied to normalize the utterance by using compensation vectors from the most similar prototype testing environment. While the compensation vectors are computed from direct frame-by-frame comparisons of speech cepstra simultaneously recorded in the training environment and various prototype testing environments, the MFCDCN algorithm does not assume that the acoustical characteristics of the actual testing environment are known. The specific compensation vector applied in a given frame depends on both the instantaneous frame SNR and the corresponding VQ codeword label.

The second algorithm, IMFCDCN, is a more general form of MFCDCN in that the compensation vectors used to normalize testing utterances are re-estimated by linear interpolation of the prototype testing environments on a sentence-by-sentence basis. In applications where the actual testing environment does not closely resemble any single prototype environments with pre-computed compensation vectors, IMFCDCN will generally provide better recognition accuracy than MFCDCN.

Chapter 6
Phone-Dependent Cepstral Normalization

So far, all compensation algorithms for unknown acoustical environments can be viewed as "signal enhancers" based on signal processing. Specifically, normalization to reduce the joint effect of noise and linear filtering due to channel changes takes place in the front end before extracted speech features are sent to the recognizer. An advantage of this approach is that the configuration of the recognizer is the same, regardless of whether compensation is conducted or not. For a complicated speech recognition system like SPHINX-II, this provides simplicity and consistency for various front ends. The decoding algorithm for speech recognition and environmental normalization procedures in the front end can be implemented and optimized independently. Any modification of the front end can be incorporated and passed on to the recognizer without extra effort. Similarly, any configuration changes in the recognition system can still make use of normalization algorithms without further modification.

On the other hand, these "signal-enhancing" compensation algorithms have some disadvantages. First, the compensation procedures only pass the best choice of compensated features to the recognizer. Other information generated during normalization is discarded once the normalization process is completed. In some situations, these compensation algorithms may make an incorrect decision and employ inappropriate compensation vectors. Therefore, incorrect normalization will adversely affect the performance of decoder. Second, these signal-enhancing compensations are derived from some criterion that is based on acoustical characteristics, such as VQ distortion from different codebooks, or some deterministic feature such as instantaneous SNR. Within this framework there is no feedback from the decoder during search that might guide further compensation activity. Similarly, information from language models is not available to help the compensation vectors.

In this chapter, we present a family of new algorithms that can be referred to as "search-based" normalization to compensate for acoustical mismatches, in contrast to the signal-enhancing techniques described in previous chapters. It will become clear that these search-based procedures operate on basically the same philosophy in addressing the issue of environmental adaptation. We anticipate that the decoder can accomplish the compensation with information from the acoustical models as well as from language models during the search when all possible choices of compensation are available.

6.1. Phone-Dependent Cepstral Normalization (PDCN)

Phonemes are the basic unit used in many existing speech recognition system such as SPHINX-II. The speech recognition system recognizes words using a recognition unit of either context-independent or context-dependent phones. The triphones used in SPHINX-II to capture intra-word contextual effects are examples of context-dependent phones. For channel normalization in a speech recognition system, the PDCN algorithm makes use of phoneme identity to classify mismatches between training and testing conditions.

6.1.1. Introduction

One might ask if the introduction of phone dependency could improve system performance more than the use of other acoustical features such as SNR or VQ codeword location. A recent experiment conducted at CMU on the AN4 task by Moreno [75] may shed some light on this. Moreno found that the recognition accuracy for PDCN with externally-provided correct phoneme identities produced a recognition accuracy on the AN4 task of 80.2%, which is increased to 85.1% when the compensation vectors are partitioned according to SNR as well as according to phoneme identity. This is virtually the level of 86.0% recognition accuracy obtained using the closetalking Sennheiser microphone, and is substantially better than the 78.6% that is observed using a conventional implementation of MDCDCN. These levels of performance are unrealistic in that correct phonemic transcriptions are not normally available to the recognition system. Nevertheless, the experiment indicates that phoneme identity can be extremely useful in compensating for mismatched acoustical environments. In related work, Beattie and Young [8] reported performance improvement by using a state-based noise-cancellation technique for noisy data collected in a moving vehicle at high speeds.

As results from the experiments described above suggest, phone-dependency may be a promising key to the compensation of environmental variability. Nevertheless, there are several issues to be resolved in order to use phoneme information for compensation. The first issue is the uncertainty of correct phonetic identities before the recognition process starts. Figure 6-1 illustrates all possible compensated outputs prior to the search process using various phone-dependent compensation vectors for a sentence of T frames long. At each time frame (vertical column), there are P

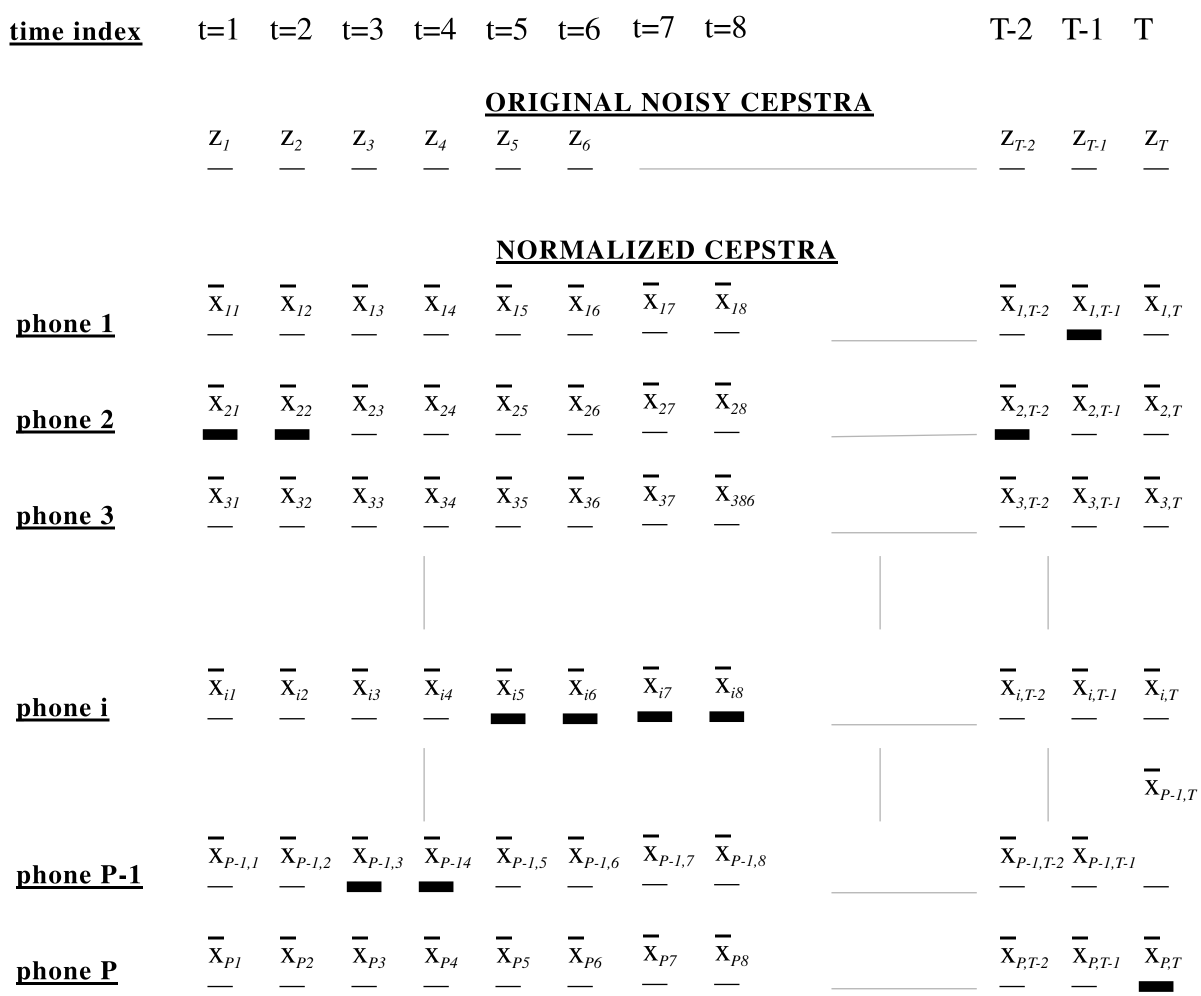


Figure 6-1. Ensemble of possible normalized outputs from the viewpoint of phone-dependent compensation. Z_t represents the original (uncompensated) cepstral vector at time *t*. $\overline{X}_{pt}$ represents the compensated output vectors at time *t* if the presumed phonetic identity is *p*. For each time frame (vertically), there are "*P*" possible compensated outputs, one for each presumed phone. The right compensated sequence illustrated by wider bars is one of the possible combinations. Note there is only one wider bar, the right compensated output, at each time frame.

choices of compensated outputs, one for each phonetic label, where *P* is the number of phonetic labels. We assume that a correct compensated output exists for every time frame and that it is the normalized output using the compensation vector of the actual phonetic identity. Correctly compensated outputs are illustrated by darker dashes in Figure 6-1. The desired compensated utterance is represented by a sequence of darker dashes. Note that there is only one dark dash (representing the correct compensation) in each vertical column. Unfortunately, the phoneme identity is not available prior to the search process. This is very different from signal-enhancing normalization algorithms in which compensation can be determined using specific deterministic acoustical characteristics such as instantaneous SNR and VQ codeword location.

One straightforward solution to combat the lack of deterministic knowledge of phonetic identity would be to perform an exhaustive search of every possible normalized output. However, the computational cost of such a procedure would be prohibitive for practical applications. In the next sections, we describe an approach developed for use within the SPHINX-II recognition system framework to provide solutions to the problems discussed above.

6.1.2. Estimation of PDCN Compensation Vectors

Given clean-speech models and a set of stereo-recorded adaptation sentences for every prototype environment, the training procedures to estimate compensation vectors of the phone-dependent cepstral normalization algorithm (PDCN) are described in Figure 6-2.

A base phone set of 51 phonemes is used in our current implementation, which includes the silence phone but excludes all other non-lexical phones due to a lack of a sufficiently large number of training samples. To enable fair comparisons, we employ the same stereo sentences to estimate compensation vectors for PDCN as had been used previously for MFCDCN in the various prototype environments. For each prototype environment, clean-speech models are used to partition data from the "clean" training microphone into phonetic segments in supervised mode. The compensation vectors are computed according to Step 3 in Figure 6-2.

Figure 6-3 shows a number of PDCN compensation vectors across different phonetic events for the PCC160 unidirectional desktop microphone. The curves in each panel show spectral profiles of compensation vectors grouped for four different types of phonemes: front vowels, back vowels, voiced fricatives, and voiced stops.

The compensation vectors for vowels exhibit the spectral profiles with decreasing magnitude for higher frequency components. In contrast, the compensation vectors of consonants exhibit less spectral slope with a slightly increasing magnitude for higher frequency components. The variation in the spectral shapes manifests the intrinsic difference among various phonemes as far as environmental mismatch is concerned. Second, the profile variation between the phones within the same group is smaller whereas the variation between the phones across different groups is larger. Third, although we only show illustrations for these four phonetic groups here, the two observations de-

1. Clean-speech models divide clean utterances into phonetic segments. Data from prototype noisy environments are labeled with the same phonetic labels obtained using clean speech. There are totally 51 phonetic labels used for the segmentation purpose.

2. For every phonetic label, a difference vector is computed by accumulating the cepstral difference between the clean training data, $\boldsymbol{x}_t$, and its noisy counterpart, z_t, in the stereo data that correspond to this particular label.

3. A compensation vector is computed by accumulating the corresponding difference from all the training utterances and then averaging as following,

$$\boldsymbol{c}[p] = \frac{\sum_{u=1}^{A} \sum_{t=1}^{T_u} (\boldsymbol{x}_{t,u} - \boldsymbol{z}_{t,u}) \delta(f_t - p)}{\sum_{u=1}^{A} \sum_{t=1}^{T_u} \delta(f_t - p)}$$

   where f_t is the phonetic label for frame t, and T_u is length of the uth utterance out of A sentences from this given prototype environment.

4. If all the prototype testing environments are processed, exit this training procedure. Otherwise go back to step 1 for the next prototype environment.

Figure 6-2. The training procedure for PDCN compensation vectors.

scribed above still hold true in other phonetic labels. Fourth, when there is no environmental mismatch, the compensation vectors will lie on the x-axis.

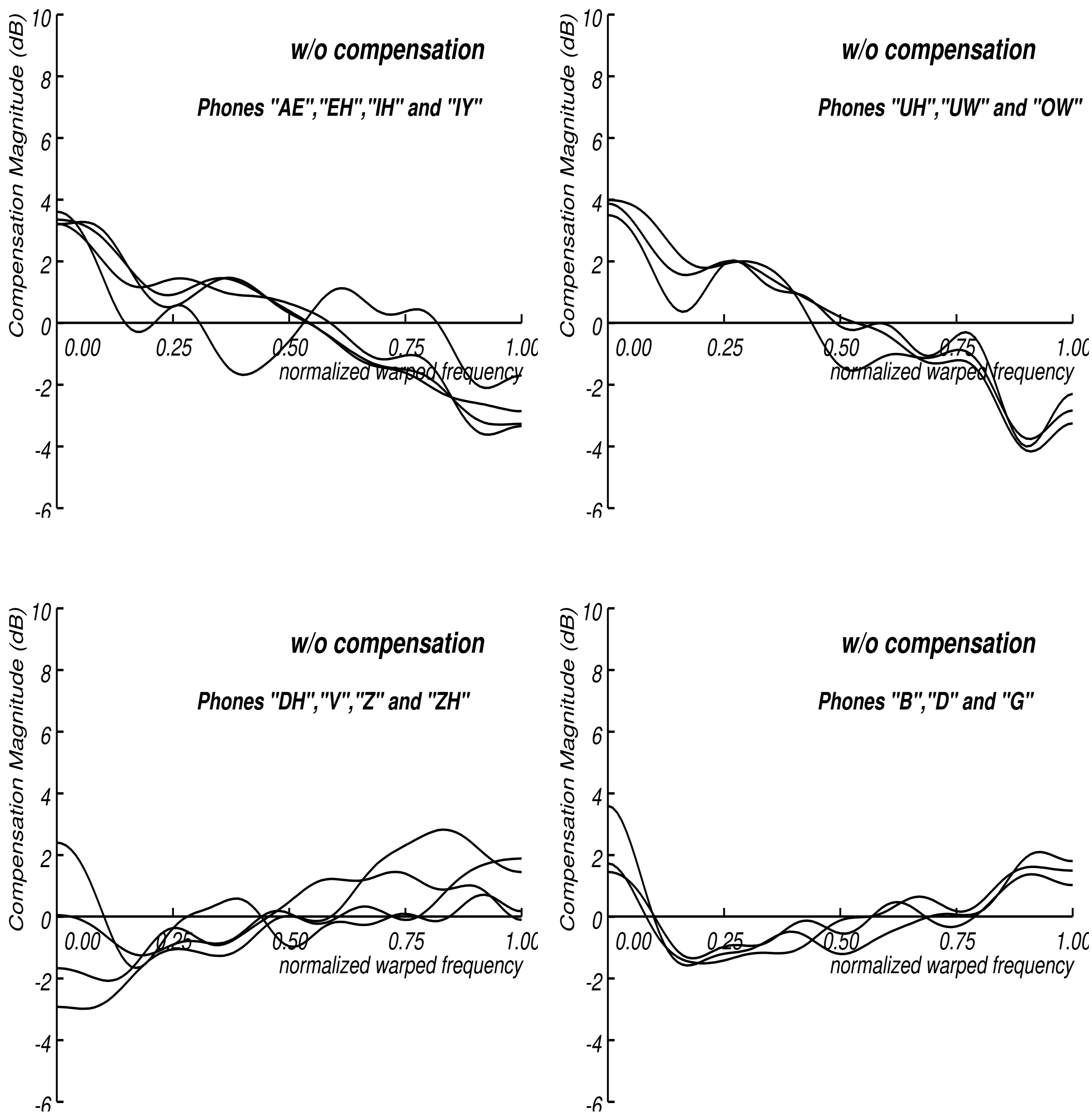


Figure 6-3. Compensation vectors of PDCN for the PCC-160 unidirectional desktop microphone. The curve in each panel reflects the differences of channel mismatch for distinctive phonemes. The four panels are for "front" vowels (the upper left panel), "back" vowels (the upper right), voiced fricatives (the lower left), and voiced stops (the lower right), respectively.

6.1.3. Application of PDCN in Testing

The SPHINX-II system uses the senone [41,42], a generalized state-based probability density function, as the basic unit to compute likelihoods from acoustical models. The probability density function for senone s for the cepstral vector $\mathbf{z}_t$ at frame t of incoming speech can be expressed as

$$Pr(\mathbf{z}_t | s) = p_{s_{\mathbf{z}_t}} = \sum_{m_z = 1}^{B} w_{m_z} \mathcal{N}(\mathbf{z}_t; \mu_{m_z}, \sigma_{m_z}) \tag{6.1}$$

where m_z stands for the index of the best B Gaussian mixtures, $\mathcal{N}()$, of senone s for frame t, and μ_{m_z}, σ_{m_z}, and w_{m_z} are the corresponding mean, standard deviation, and weight for the m_z^{th} mixture of senone s.

To compensate for environmental mismatches, the PDCN compensation vectors obtained during the estimation phase described above are applied during recognition. PDCN aims to eliminate the effects of changing conditions in terms of presumed phonetic identity. Multiple compensated cepstral vectors are formed in PDCN by adding on a frame-by-frame basis the original compensation vectors to incoming cepstra, $\hat{\boldsymbol{x}}_{t,p}(t)$, where, $\hat{\boldsymbol{x}}_{t,p} = z_t + \mathbf{c}[p]$. Nevertheless, the correctly compensated output still needs to be determined.

Because a senone in the SPHINX-II system is shared by triphones that correspond to the same base phoneme, senones are identified according to the phonetical label of the corresponding distinct base phone. When a specific senone probability is to be computed, it employs the PDCN normalized output that corresponds to that particular base phoneme. Thus, the senone probability with PDCN is re-written as

$$p_{s_{\hat{x}_t}} = \sum_{m_{\hat{x}} = 1}^{B} w_{m_{\hat{x}}} \mathcal{N}\left(\hat{\mathbf{x}}_{t,p_s}; \mu_{m_{\hat{x}}}, \sigma_{m_{\hat{x}}}\right) \tag{6.2}$$

where $m_{\hat{x}}$ is the index of the best B Gaussian mixtures for senone s at frame t with respect to the PDCN-normalized cepstral vector $\hat{\boldsymbol{x}}_{t,p_s}$, for the corresponding phonetic label for senone s.

During the search process, the optimal search path is primarily determined by the acoustical models and language models. Therefore, PDCN compensates for the effect of linear filtering in the way that the correct compensation vector is implicitly chosen inside the decoder. The overall score used in the recognition process is derived from the cepstral vectors used for the final recognition result, *after* the environmental compensation. Note that only one probability calculation is carried out for each senone in each time frame, the same as with a baseline system. The increase in computational cost of PDCN relative to the baseline is minor, and it comes primarily from the addition of cepstral compensation and vector quantization.

Processing Algorithm	**CMN**	**CMN +PDCN**	**CMN +MFCDCN**
CLSTK (Training Mic)	7.6	7.9	7.6
Error Reduction	–	–3.9	0.0
Secondary-Mic Data	21.4	16.9	14.5
Error Reduction	–	21.0	32.2

Table 6-1. Percentage of word errors and corresponding error rate reduction for PDCN in combination with CMN on the ARPA WSJ0-si_evl5 task.

Table 6-1 compares recognition word error rates for the ARPA WSJ0-si_evl5 task using cepstral mean normalization (CMN) only, the PDCN algorithm with CMN (CMN+PDCN), and the MFCDCN algorithm with CMN (CMN+MFCDCN). The PDCN algorithm, combined with cepstral mean normalization, generates a 21.0% reduction in word error rate in comparison with results using CMN alone. This indicates that the PDCN algorithm is able to provide environmental normalization that reduces the error rate substantially over the baseline when there is a mismatch of acoustic environments.

However, there are several issues that need to be noted. First, we anticipated that PDCN could take advantage of the discriminating ability of the decoder to determine and choose the correct compensation from the set of possible compensated vectors. Yet we are disappointed to find that the performance improvement of the PDCN algorithm does not exceed that of MFCDCN. Second, due to the complexity of computing delta features over a segments that include phoneme transitions, PDCN provides compensation only for the static cepstral vectors and not the delta cepstra and delta-delta cepstra). The complexity arises from the fact that the derivation of dynamic features depends on past data as well as on future data. In contrast, typical signal-enhancing algorithms compensate for mismatches on the static features first, and then derive other dynamic features from the compensated data before they are input to the decoder.

We will address these issues in details in the following sections. New approaches and extensions are developed to investigate these considerations.

6.2. Combination of MFCDCN and PDCN

In the previous section, it was shown that PDCN is quite useful in combating the mismatch problem for unknown acoustical testing environments. Nevertheless, we do notice that PDCN still needs to improve in order to decrease the gap between its accuracy and that of a baseline system using clean speech. As noted above, a problem with PDCN is that the selection of proper phone-dependent compensation vectors is complicated by decoder errors in determining the exact phoneme sequences. It is our hypothesis that misrecognition owing to the decoder's potentially poor selection of compensation vectors for noisy speech can be reduced with "cleaner" testing speech data from unknown acoustic environments. Because PDCN compensation takes place during the search, it is not difficult to apply some signal-enhancing techniques to improve the quality of speech before recognition.

Figure 6-4 shows one approach to combining PDCN with a signal-enhancing compensation algorithm. In general, any effective signal-enhancing compensation can be used. In this dissertation, we use MFCDCN because of its ability to provide substantial robustness to unknown testing environments. The training process to estimate compensation vectors in this approach is the same as in PDCN, except that noisy speech from every target environment needs to be processed first using MFCDCN. Similarly, testing utterances are also enhanced using MFCDCN before recognition. The PDCN algorithm is, then, applied to provide further compensation for possible mismatches during the search process.

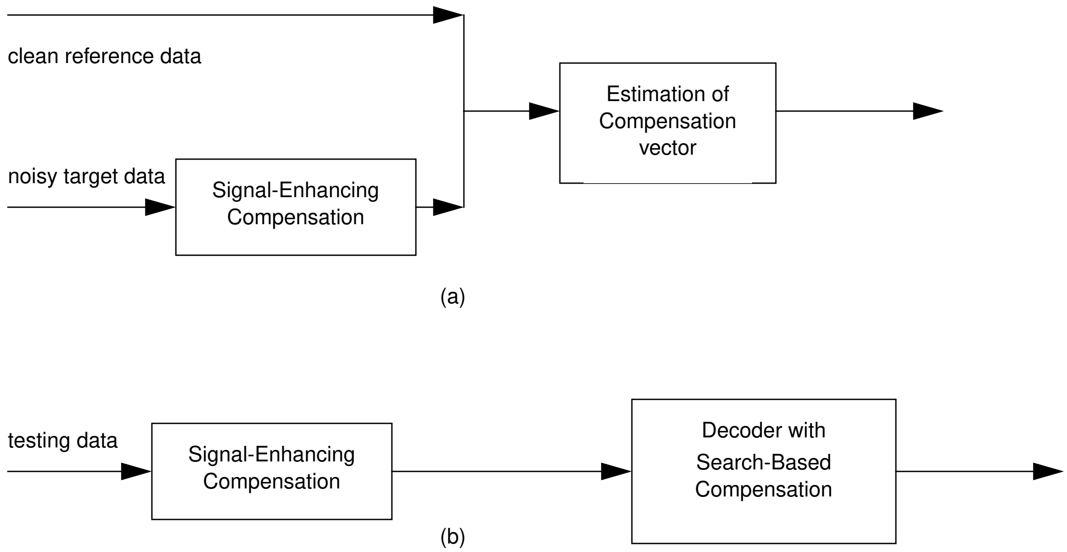


Figure 6-4. Block diagram of the recognition system with compensation in both search-based and signal-enhancing compensation. In this section, PDCN is used as a search-based compensation and MFCDCN is used as the signal-enhancing compensation. (a) illustrates the application of the signal-enhancing compensation to the noisy speech to estimate compensation vectors for PDCN in the training phase. (b) shows compensation using both signal-enhancing and search-based compensation for each testing sentence.

Figure 6-5 shows a set of PDCN compensation vectors for the unidirectional desktop PCC-160 microphone, obtained in combination with MFCDCN. The compensation vectors are illustrated for the same four different phonetic groups as in Figure 6-3: front vowels, back vowels, voiced fricatives, and voiced stops. Because of the prior compensation provided by MFCDCN, the compensation vectors in Figure 6-5 exhibit less spectral variation between the training data and transformed testing data than the compensation vectors in Figure 6-3.

If the compensated testing speech were identical to the training speech, the profile would lie on the x-axis, an upper limit for compensation vectors. Figure 6-5 also demonstrates that MFCDCN reduces the spectral variability for these four phonetic groups in that they are closer to the x-axis than the curves of Figure 6-3. We also note that the reduction of spectral variation is greater for vowels than for consonants.

To provide a more useful view of several selected phonemes, we show in Figure 6-6 individual comparisons of compensation vectors for the phonemes, “AE”, “D”, and “SIL”. For the original (unnormalized) testing speech, changes in environment produce more obvious variability for vowels than for consonants. The silence segments of speech also exhibit noticeable spectral variability.

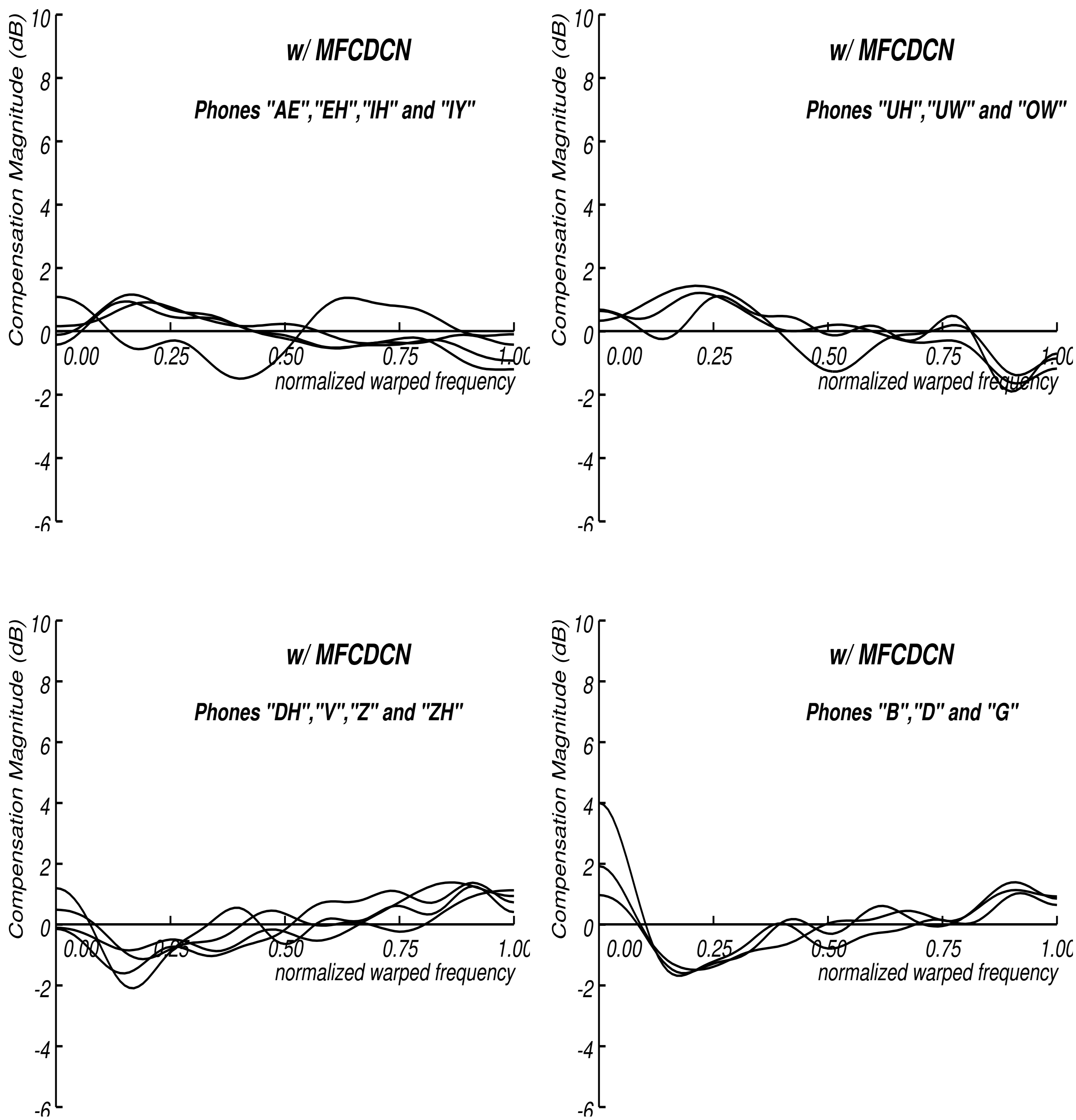


Figure 6-5. The comparison of compensation vectors of PDCN for the PCC-160 unidirectional desktop microphone with data normalized by MFCDCN before the estimation. The four panels are for “front” vowels (the upper left panel), “back” vowels (the upper right), voiced fricatives (the lower left), and voiced stops (the lower right), respectively.

However, after MFCDCN is applied, the spectral profiles are moved more closely to the x-axis. This demonstrates clearly the reduction of environmental mismatches due to the application of MFCDCN.

Processing Algorithm	CMN	CMN +PDCN	CMN +MFCDCN	CMN+ +MFCDCN +PDCN
CLSTK (Training Mic)	7.6	7.9	7.6	7.6
Error Reduction	–	–3.9	0.0	0.0
Secondary-Mic Data	21.4	16.9	14.5	12.9
Error Reduction	–	21.0	32.2	39.7

Table 6-2. Comparison of results for PDCN in different combinations, as well as with/without MFCDCN, using the ARPA WSJ0-si_evl5 task

Table 6-2 compares results obtained by combining the PDCN algorithm with the MFCDCN algorithm in differing ways. When combined with MFCDCN, PDCN can reduce the word error rate to 12.9%, a 40% error rate reduction over the CMN result. Compared to results obtained using CMN and MFCDCN alone, the combination of CMN, MFCDCN, and PDCN provides an addition error reduction of 11.0% as the error rate is reduced from 14.5% to 12.9%. By the same token, compared to the result obtained using CMN and PDCN alone, MFCDCN reduces the error rate from 16.9% to 12.9%, equivalent to an error rate reduction of 23.7%.

The complementary improvement obtained from combining MFCDCN with PDCN agrees with our hypothesis that the discriminative power of PDCN is constrained by internal errors caused by changes of acoustic characteristics in testing environment. It also demonstrates the feasibility

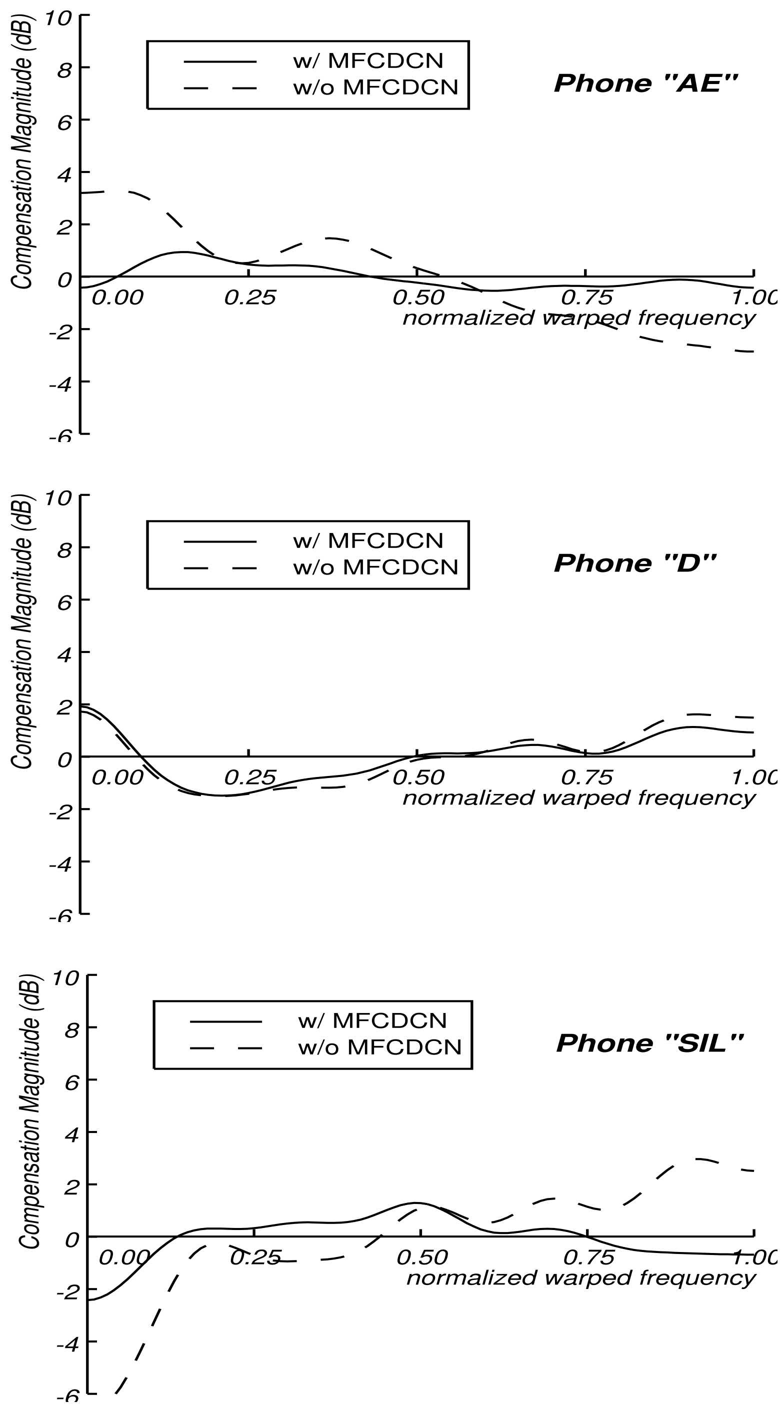


Figure 6-6. Comparison of PDCN compensation vectors with and without MFCDCN as front-end compensation. The power component, c[0], is not included in these figures. The compensation vectors are computed with the PCC-160 unidirectional desktop microphone data for three different phonemes “AE”, “D”, and “SIL”.

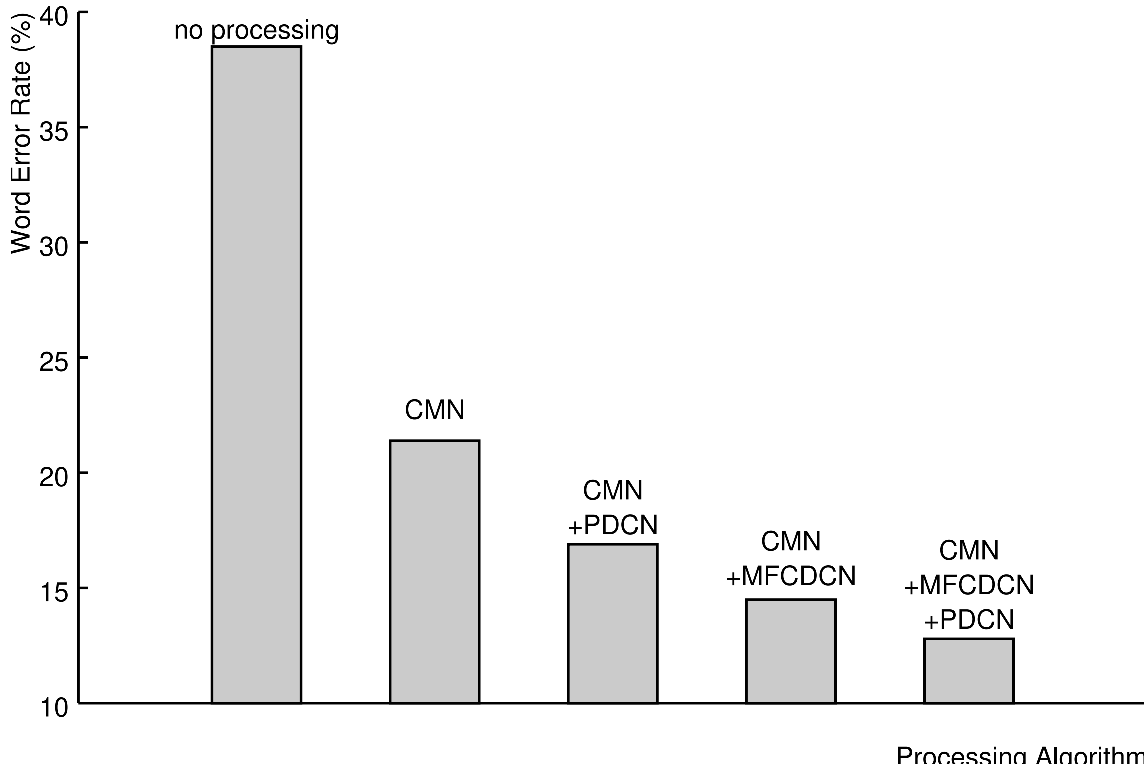


Figure 6-7. Word error rates for the secondary-microphone data from the ARPA WSJ0-si_evl5 task.

of using a combination of signal-enhancing compensation and search-based compensation. Figure 6-7 summarizes these results in graphical form.

6.3. Interpolated Phone Dependent Cepstral Normalization (IPDCN)

As described in the previous chapter, PDCN exploits compensation from a single environment in the prototype testing set that is believed to have acoustical characteristics most resembling those of the incoming testing environment. In some cases, the incoming test environment does not resemble any environment in the prototype testing set. In these cases, interpolating the compensation vectors of several environments may be helpful. Therefore, we present an extension of PDCN, referred to as Interpolated Phone Dependent Cepstral Normalization (IPDCN), to cope with this issue.

6.3.1. Gaussian Classification

In the previous chapter we discussed two methods of environment selection: classification by compensation and classification based on Gaussian features. Due to the complexity of utilizing compensated output for environment selection during the search, we choose to employ the Gaussian environment classifier [65] described in Section 5.3.3.. This procedure models each prototype environment with mixtures of Gaussian density functions. Environment selection is accomplished so that the original (uncompensated) test data has the greatest probability from the corresponding classifier.

The weighting factors used in IPDCN for each environment for an utterance $\bar{\mathbf{Z}}=\mathbf{z}_1\mathbf{z}_2...\mathbf{z}_t..\mathbf{z}_T$ are proportional to the probability of environment given the observed data as shown in Equation (5.8) and Equation (5.9). As each prototype environment is modeled as mixtures of Gaussian density functions for the original data, the probability of cepstral vector, z_t, given environment e is defined as

$$p(z_t|e) = \sum_{m_z=1}^{B} \frac{C}{\sigma_{m_z,e}} exp\{\|z_t - \mu_{m_z,e}\|^2 / (-2\sigma^2_{m_z,e})\} \tag{6.3}$$

where $\mu_{m_z,e}, \sigma_{m_z,e}$ are the mean and variance of the m_z^{th} "best" mixture of Gaussian density from the e^{th} environment among all E prototype testing environments with respect to the cepstral vector $\mathbf{z}_t$, and B is the number of Gaussian mixtures to be considered.

The probability of observing the utterance, $\bar{\mathbf{Z}} = z_1 z_2 z_3 ...... z_t ..... z_T$, given environment e can be expressed as

$$p(\bar{\mathbf{Z}}|e) = \prod_t^T p(z_t|e) = \prod_t^T \sum_{m_z=1}^{B} \frac{C}{\sigma_{m_z,e}} exp\{\|z_t - \mu_{m_z,e}\|^2 / (-2\sigma^2_{m_z,e})\} \tag{6.4}$$

The corresponding weight factors are

$$f_e = \frac{p(e|\bar{\mathbf{Z}})}{\sum_{i=1}^{E} p(i|\bar{\mathbf{Z}})} = \frac{\prod_{t=1}^{T} \sum_{m_z=1}^{B} \frac{1}{\sigma_{m_z,e}} exp\{\|z_t - \mu_{m_z,e}\|^2 / (-2\sigma^2_{m_z,e})\}}{\sum_{i=1}^{E} \prod_{t=1}^{T} \sum_{m_z=1}^{B} \frac{1}{\sigma_{m_z,i}} exp\{\|z_t - \mu_{m_z,i}\|^2 / (-2\sigma^2_{m_z,i})\}} \tag{6.5}$$

6.3.2. Application of IPDCN in Testing

The compensation vectors to be used in IPDCN are estimated for each of the prototype testing environments by using the standard PDCN training algorithm described in Section 6.1.2.. In the recognition phase, IPDCN computes the weighting factor for each environment as in Equation (6.5). For each testing sentence, compensation vectors are obtained by linear interpolation of several PDCN compensation vectors precomputed for all prototype environments as

$$\hat{\boldsymbol{c}}[p] = \sum_{e=1}^{E} f_e \cdot \boldsymbol{c}[p, e] \tag{6.6}$$

where $\hat{\boldsymbol{c}}[p]$, $\boldsymbol{c}[p, e]$, *and* f_e are the estimated compensation vectors, the environment-specific compensation vector for the e^{th} environment, and the weighting factor for the e^{th} environment, respectively.

After the sentence-based interpolated compensation vectors are obtained, they are applied for environmental normalization during the search in the same manner as the PDCN algorithm described in Section 6.1.3. In the current implementation of IPDCN, we use the 3 closest environments with the best 4 Gaussian mixtures in interpolation. It can be seen that PDCN is equivalent to IPDCN with $E = 1$.

Table 6-3 and Table 6-4 list results obtained using IPDCN as well as similar results using PDCN and MFCDCN in conjunction with cepstral mean normalization (CMN) using the APRA WSJ0-si_evl5 set. The difference between these two experiments is that results in Table 6-3 were obtained with all three testing environments included among the prototype environments, while the results of Table 6-4 were obtained when the testing environments were excluded from the prototype environments. We summarize these results as follows: (1) IPDCN exhibits a slight performance improvement with respect to PDCN when the test environments are included in the prototype environments. (2) As we expect, error rates increase when the testing environment is not included in the set of prototype environments. Specifically, the system using PDCN+MFCDCN+CMN exhibits a 14.7% error-rate increase from 12.9% to 14.8% when the testing microphone is taken out of the training corpus. (3) It is shown in Table 6-4 that IPDCN provides only slight improvement relative to PDCN when the correct compensation vectors are not available.

Processing Algorithm	**CMN**	**CMN +PDCN**	**CMN +IPDCN**	**CMN +MFCDCN +PDCN**	**CMN +MFCDCN +IPDCN**
CLSTK (Training Mic)	7.6	7.9	7.7	7.6	7.6
Error Reduction	–	–3.9	–1.3	0.0	0.0
Secondary-Mic Data	21.4	16.9	16.5	12.9	12.3
Error Reduction	–	21.0	22.9	39.7	42.5

Table 6-3. Comparison of word errors and corresponding error rate reduction of IPDCN with top 3 prototype testing environments (*E*=3) in conjunction with CMN on the ARPA WSJ0-si_evl5 task

Processing Algorithm	**CMN**	**CMN +MFCDCN +PDCN**	**CMN +MFCDCN +IPDCN**	**CMN +IMFCDCN +PDCN**	**CMN +IMFCDCN +IPDCN**
CLSTK Training Mic	7.6	7.6	7.6	7.6	7.6
Error Reduction	--	0.0	0.0	0.0	0.0
Secondary-Mic Data	21.4	14.8	14.7	13.5	13.5
Error Reduction	--	30.8	31.3	36.9	36.9

Table 6-4. Recognition accuracy obtained for the same task as in Table 6-3, but with all three test environments excluded from the list of prototype environments.

6.4. Other Considerations

As described in previous sections, the approach of PDCN appears to be effective in dealing with the issue of environmental mismatches. In this section, we investigate two additional procedures that improve the effectiveness of PDCN. The first procedure extends PDCN by including the instantaneous frame SNR in selecting compensation vectors. The second approach examines how PDCN may be applied to other speech features used in the recognition system besides the static cepstral coefficients.

6.4.1. Use of SNR information in PDCN

The performance improvement provided by BSDCN suggests a simple extension of PDCN to incorporate instantaneous SNR in partitioning the spectral space for the environmental compensation vectors. We use the initials SPDCN to designate the PDCN algorithm augmented by further partitioning of the feature space according to SNR.

The training process of SPDCN is the same as for PDCN except that the instantaneous frame SNR needs to be considered in addition to presumed phonetic labels from the sentence segmentation process, as shown in Equation (6.7).

$$\boldsymbol{c}[p, l] = \frac{\sum_{u=1}^{A} \sum_{t=1}^{T_u} (\boldsymbol{x}_t - \boldsymbol{z}_t)\, \delta(f_t - p)\, \delta(s_t - l)}{\sum_{u=1}^{A} \sum_{t=1}^{T_u} \delta(f_t - p)\, \delta(s_t - l)} \tag{6.7}$$

where s_t is the instantaneous frame SNR of z_t at time frame t, f_t is the phoneme, and T_u is length of the u^{th} utterance out of A sentences from each prototype environment.

In the recognition phase, the instantaneous frame SNR is also computed for determining which compensation vectors should be tried during the search. As in PDCN, at each time frame multiple compensated cepstral vectors are formed by adding various compensation vectors to the incoming cepstra given the instantaneous SNR, $\hat{\mathbf{x}}_{t,p} = \mathbf{z}_t + \mathbf{c}[p, s_t]$, on a frame by frame basis. The senone probability density function is the same as described in Equation (6.2). Environment interpolation can be applied to SPDCN in the same manner as IPDCN using Equation (6.5) and Equation (6.6).

Processing Algorithm	**CMN**	**CMN +MFCDCN**	**CMN +PDCN**	**CMN +SPDCN**
CLSTK (Training Mic)	7.6	7.6	7.9	7.6
Error Reduction	–	0.0	–3.9	0.0
Secondary-Mic Data	21.4	14.5	16.9	15.9
Error Reduction	–	32.2	21.0	25.7

Table 6-5. Result of SNR-dependent PDCN (SPDCN) on the ARPA WSJ0-si_evl task.

Processing Algorithm	**CMN**	**CMN +MFCDCN +PDCN**	**CMN +MFCDCN +IPDCN**	**CMN +MFCDCN +SPDCN**	**CMN +MFCDCN +ISPDCN**
CLSTK (Training Mic)	7.6	7.6	7.6	7.6	7.6
Error Reduction	–	0.0	0.0	0.0	0.0
Secondary-Mic Data	21.4	12.9	12.3	12.7	12.3
Error Reduction	–	39.7	42.5	40.7	42.5

Table 6-6. Comparisons for SNR-dependent PDCN as well as Interpolated SPDCN (ISPDCN) in conjunction with CMN on the ARPA WSJ0-si_evl5 task.

Table 6-5 and Table 6-6 compare results obtained using SPDCN and/or Interpolated SPDCN. The incorporation of frame-SNR information in PDCN reduces the error rate from 16.9% to 15.9% for the system without MFCDCN in Table 6-5. Meantime, Table 6-6 shows only a marginal improvement from 12.9% to 12.7% produced by the inclusion of frame-SNR information in the compensation process in combination with MFCDCN. The lack of greater benefit observed in Table 6-6 may be attributable to information redundancy in the MFCDCN vectors. Because vowels usually exhibit higher SNRs and consonants exhibit lower SNR, phonetic labels and SNRs are likely to be highly correlated. Therefore, the possible benefit to be obtained from combining SNRs with phonetic labels may be limited. Besides, the system has also benefited some significant compensations from the well-estimated MFCDCN vectors.

6.4.2. Compensation of Cepstral Differenced Vectors

The SPHINX-II system employs four features, cepstrum, differenced cepstrum, second-order differenced cepstrum, and power vectors, to compare the acoustic input to the phonetic models. For a typical signal-enhancing compensation technique, the dynamic features (differenced cepstrum, second-order differenced cepstrum, and differenced power) are derived by computing the corresponding differences of compensated static cepstra.

However, the PDCN algorithm and its extensions operate only on static cepstra due to the complexity of computing other dynamic features over a window of normalized cepstra during the search. Specifically, the derivation of dynamic features depends on the past normalized cepstra as well as on future normalized cepstra. The need for both past and future normalized outputs makes the derivative features difficult to obtain during the search process. As noted above, it is generally believed [35, 77] that the derivative cepstral features are less susceptible to steady-state and slow-moving variations introduced by varying the microphone or channel characteristics.

In an effort to shed some light on the potential improvement that could be provided by dynamic features, we conducted a small number of experiments in which compensated dynamic features were computed during the search process, using the procedure summarized in Figure 6-8. For simplicity, we assume that the effects of environmental variabilities on one feature are independent of those of another, and that they can be identified by the phonetic labels of current time frames. Thus, compensation can be applied to each feature individually using compensation vectors based on the presumed phonetic label.

As with the training procedure described in Section 6.1., phonetic information for all utterances is obtained by applying the clean-speech models to divide the clean speech into phonetic segments. Then, for each of the speech features, the training algorithm of PDCN described in Figure 6-2 is employed to compute compensation vectors based on the information of phonetic segmentation. This procedure is repeated for each of the prototype environments. During recognition, compensation is carried out by applying proper compensation vectors to the desired set of features.

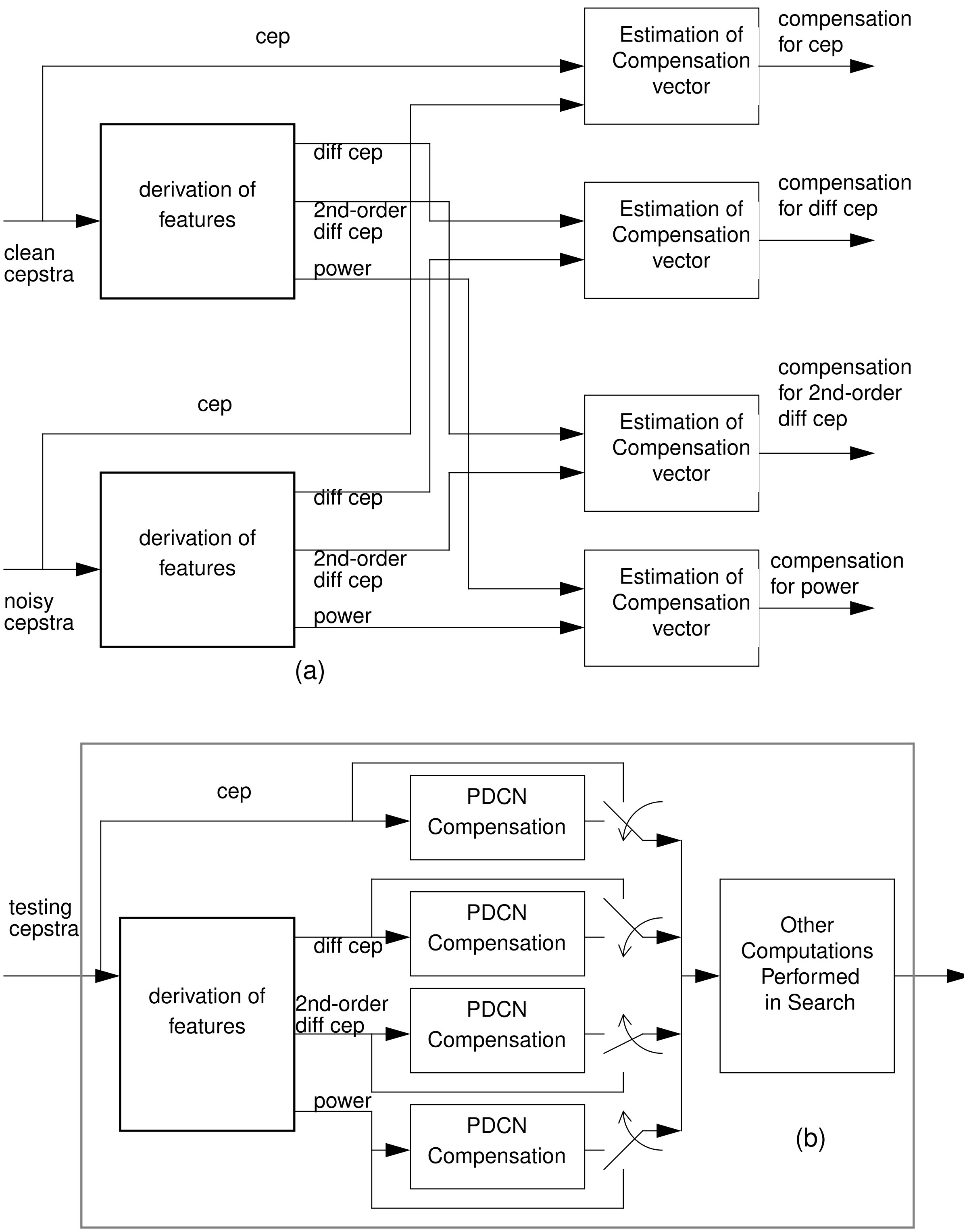


Figure 6-8. Block diagram of compensation applied to any of four features used by SPHINX-II. (a) the training phase. (b) the recognition phase.

It can be easily seen in Figure 6-8(b) that the configuration with compensation only on the static cepstral feature, “cep”, is the PDCN algorithm described in previous sections.

Features Normalized	**cep only**	**dcep only**	**xcep only**	**pcep only**	**cep+ dcep**	**cep+ xcep**	**cep+ pcep**	**cep+ xcep+ pcep**	**cep+ dcep+ xcep+ pcep**
Error Rate (2nd-Mic)	12.9	14.8	14.4	14.7	12.9	12.7	12.7	12.7	12.9

Table 6-7. Word error rates of the application of phone-dependent compensation to different features. Note MFCDCN is used also. They are obtained for the secondary-microphone data in the ARPA WSJ0-si_evl5 task. The notations indicate the compensated features, where “cep” stands for static cepstra, “dcep” for differenced cepstra, “xcep” for second-order differenced cepstra, and “pcep” for power.

Table 6-7 summarizes results obtained applying phone-dependent compensation to different features used by SPHINX-II in combination with MFCDCN. We note the following: (1) Individual compensation of features other than static cepstra does not produce substantial error reduction. (2) Application of compensation to other features in addition to static cepstra only produces marginal improvement, if any. (3) Among the three derivative features, compensation of differenced cepstra appears to provide the least help. We hypothesize that this is due to a larger window (80 msec) used to compute differenced cepstra than the other two derivative features (40 msec). These results agree with our underlying assumption of PDCN that static cepstral vectors are more susceptible to changes in acoustic environments than other derivative features. It also indicates that more sophisticated approaches need to be developed to benefit from environmental compensation of dynamic features.

6.5. Summary

In this chapter, we presented a different approach to circumventing the problem of mismatched training and testing acoustic conditions. PDCN and its extensions represent another paradigm for normalization, as compensation for environmental mismatches takes place during the search. The development of PDCN is motivated by the fact that the discriminative ability of the decoder can be used to select the compensation vectors effectively for unknown testing conditions. It can take advantage of useful information from the decoder by deferring the choice of compensation vectors until the search. It also suggests a methodology for utilizing the information from acoustic models as well as language models in providing better environmental normalization.

When applied without other environmental compensation, PDCN provides a level of environmental normalization that is comparable to that of MFCDCN. Further improvements can be achieved when PDCN is applied in combination with MFCDCN.

Using linear interpolation of the compensation vectors, IPDCN can estimate new compensation vectors for unknown testing environments that are not necessarily included in the prototype environments. Robustness may be improved by using this technique even in situations where the testing microphones are among the prototype environments in the training set.

Other extensions of PDCN are described, including the use of SNR information for PDCN and individual compensation on other features. However, only marginal benefits from these two extensions were observed.

Chapter 7
Environmental Adaptation via Codebook Adaptation

Many approaches [19, 92, 39, 48, 54, 98] have been studied for enabling speech recognition systems to accommodate different acoustical characteristics between speakers. In the domain of speaker adaption, codebook adaptation [92, 98] has been successfully applied to cope with spectral variations between training and testing speakers. In this chapter, we will investigate some ways in which codebook adaptation techniques can be applied to resolve the problem of environmental mismatches.

7.1. Introduction

A vector quantization (VQ) codebook [29] is useful in characterizing the acoustical space of speech data used in VQ clustering. The codebook generated for a given speaker or environmental condition can be used to distinguish this speaker or environment from others. Hence, the codebook is regarded as a representation of the speaker or environment and can be used in simple identification tasks. For example, a well-trained gender-specific codebook can be used to perform accurate gender classification [64].

For characterization of environmental conditions, adaptation can be used to “reshape” or “adjust” condition-specific codebooks to match the distribution of incoming speech vectors representing a new target condition. For example, codebook adaptation to new speakers is an effective way to increase recognition accuracy to the level of accuracy for a speaker-dependent system [92]. With the availability of adaptation data from the target speaker, a mapping can be established to relate the spectral space of the target speaker to that of the reference speaker via codebook adaptation.

In general, errors in the vector quantization process would be smaller if the distribution of incoming speech matches the distribution characterized by the codebook. Similarly, the application of a VQ codebook obtained for a specific condition can produce substantial quantization errors for data from mismatched conditions. This indicates that a codebook needs to be “tuned” to better characterize the spectral space of testing data. It is reasonable to anticipate that this general technique of codebook adaptation can be extended to address spectral variability due to changes in environment.

In the following sections, we will study and evaluate two techniques using codebook adaptation to accomplish robust speech recognition in the context of changes in environment, dual-channel codebook adaptation, and Baum-Welch codebook adaptation.

7.2. Dual-Channel Codebook Adaption (DCCA)

In this section, we propose the Dual-Channel Codebook Adaptation (DCCA) technique to handle environment mismatches between training and testing conditions. DCCA exploits the availability of simultaneous recordings of speech samples from two microphones, and an existing codebook in the recognition system for the training environment.

The DCCA technique depends on VQ indices to classify acoustical variabilities between training and testing environments. The original codebook to be adapted is developed using speech from a clean training environment, and it consists of mean vectors and variances used in Gaussian mixtures, and VQ label indices. We consider two different methods for dual-channel codebook adaptation. The first approach involves changing both the means and variances in the system codebook, and the second involves updating the means only while keeping variance unchanged.

7.2.1. Adaptation of the Means and Variances

The advantage of codebook adaptation for acoustical robustness is that it presents an alternative approach to environmental compensation compared to the various signal-enhancing compensation algorithms. In the SPHINX-II system with traditional signal-enhancing compensation algorithms, the senone probability density function p_{s_t} for senone s_t at time t in terms of mixture Gaussian distributions, after compensation, can be expressed as

$$p_{s_t} = \sum_{k=1}^{B} w_k \mathcal{N}(\hat{\boldsymbol{x}}_t;\ \mu_k, \sigma_k) = \sum_{k=1}^{B} w_k \mathcal{N}(z_t + \delta z_t;\ \mu_k, \sigma_k) \tag{7.1}$$

where $k, z_t, \delta z_t, \hat{\boldsymbol{x}}_t, \mu_k, \sigma_k$ are the mixture indices among the top B mixtures, the noisy observation vector, compensation vector, compensated vector, and mean vector and variance for the k^{th} Gaussian mixture, respectively. The senone probability density function p_{s_t} can be also expressed as

$$p_{s_t} = \sum_{k=1}^{B} w_k \mathcal{N}(z_t;\ \mu_k + \delta\mu_k, \sigma_k + \delta\sigma_k) = \sum_{k=1}^{B} w_k \mathcal{N}(z_t;\ \hat{\mu}_k, \hat{\sigma}_k) \tag{7.2}$$

where $\delta\mu_k$ and $\delta\sigma_k$ are the differences of the mean vector and variance in corresponding Gaussian mixtures between the noisy testing environment and the reference training environment. Equation (7.2) is a general characterization of codebook adaptation. In the issue of environmental variability, $\delta\mu_k$ and $\delta\sigma_k$ account for changes in environment. variability. It can be also seen from Equation (7.2) that in the approach of DCCA, the effects of $\delta\mu_k$ and $\delta\sigma_k$ are embedded in the transformed mean vectors and variances.

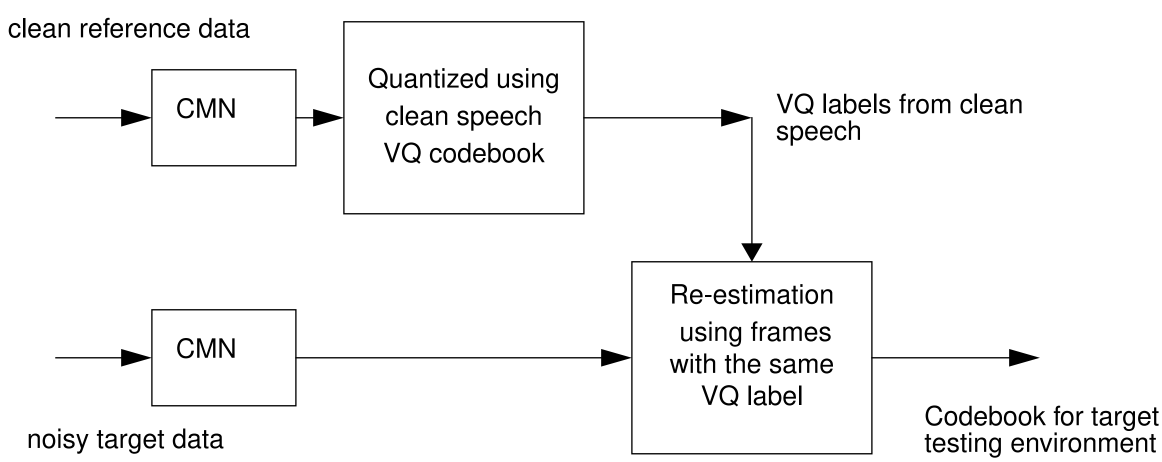


Figure 7-1. Block diagram of dual-channel codebook adaptation by using simultaneous recording data of training environment and target testing environment.

Figure 7-1 is a block diagram of the re-estimation procedure for the transformed codebook for each environment. First, VQ encoding is performed on the clean data only. The output VQ labels are to be shared by the reference training environment and the target testing environment. Note that CMN is performed on utterances from both training and testing environments. Since the original codebook is estimated with CMN-normalized training data, the transformed codebook must be developed using CMN as well.

For each subspace in the training environment, we generate the corresponding mean vectors and variances for the target testing environment as described in Equation (7.3) and Equation (7.4)

$$\hat{\mu}_k = \frac{\sum_t z_t \cdot \delta(l_t - k)}{\sum_t \delta(l_t - k)} \tag{7.3}$$

$$\hat{\Sigma}_k = \frac{\sum_t (z_t - \hat{\mu}_k)(z_t - \hat{\mu}_k)^T \cdot \delta(l_t - k)}{\sum_t \delta(l_t - k)} \tag{7.4}$$

Thus, a one-to-one mapping between training and target condition is established in terms of mean vectors and variances. This maintains the validity and integrity between the original (un-adapted) output probabilities and the updated codebooks.

Table 7-1 shows the performance of DCCA on the ARPA WSJ0-si_evl5 task. For DCCA, the Gaussian environment classifier described in Section 5.3.3. is employed to determine the appropriate codebooks. Combined with CMN, DCCA achieves a word error rate of 14.9%, comparable to that of MFCDCN. This result indicates that DCCA is quite effective in compensating for environmental mismatches. We also combined DCCA with MFCDCN. As shown in Table 7-1, a further improvement can be obtained when the word rate is reduced from 14.9% to 13.5%.

7.2.2. Adaptation of the Means Only

The goal of signal-enhancing compensation algorithms is to transform the noisy speech signal to match the spectral space of the training signal. The normalized cepstra are to be evaluated using the same system parameters, such as the means, variances and senones, estimated based on the clean training database as described in Equation (7.5).

$$p_{s_t} = \sum_{k=1}^{B} w_k \mathcal{N}(\hat{x}_t; \mu_k, \sigma_k) = \sum_{k=1}^{B} w_k \mathcal{N}(z_t + \delta z_t; \mu_k, \sigma_k) \tag{7.5}$$

It is worth noting that this kind of approach assumes that the acoustical space of normalized cepstra can match that of the training cepstra, and therefore these system parameters can be directly applied to the normalized cepstra.

Processing Algorithm	CMN	CMN+ MFCDCN	CMN+ DCCA	CMN+ MFCDCN+ DCCA
CLSTK (Training Mic)	7.6	7.6	7.8	7.5
Error Reduction	--	0.0	-2.6	1.3
Secondary-Mic Data	21.4	14.5	14.9	13.5
Error Reduction	--	32.2	30.4	36.9

Table 7-1. Percentage of word errors and corresponding error rate reduction for dual-channel codebook adaptation (DCAA) for the ARPA WSJ0-si_evl5 test data. In this experiment, mean vectors as well as variances of the Gaussian mixtures are re-estimated.

From the viewpoint of the recognition system, correction based on signal-enhancing algorithms exhibits the equivalent effect of shifting the mean vectors of Gaussian mixtures while the variances are kept the same. The senone probability density function p_{s_t} for senone s_t at time t can be re-expressed as in Equation (7.6).

$$p_{s_t} = \sum_{k=1}^{B} w_k \mathcal{N}(z_t; \mu_k - \delta z_t, \sigma_k) \tag{7.6}$$

An examination of Equation (7.6) suggests an alternative form of codebook adaptation, implemented by tuning only the mean vectors while the variances are kept the same as those of the clean data. Figure 7-2 illustrates the effect of codebook adaptation on the Gaussian mixtures when the means are updated but variances remained unchanged. In other words, codebook adaptation is accomplished using the procedure described in Figure 7-1, except that the variances are not updated. In the DCCA approach with updating means only, referred to as DCCA2, the senone probability density function p_{s_t} can be described by Equation (7.7).

$$p_{s_t} = \sum_{k=1}^{B} w'_k \mathcal{N}(z_t;\ \mu_k + \delta\mu_k, \sigma_k) = \sum_{k=1}^{B} w'_k \mathcal{N}(z_t;\ \hat{\mu}_k, \sigma_k) \tag{7.7}$$

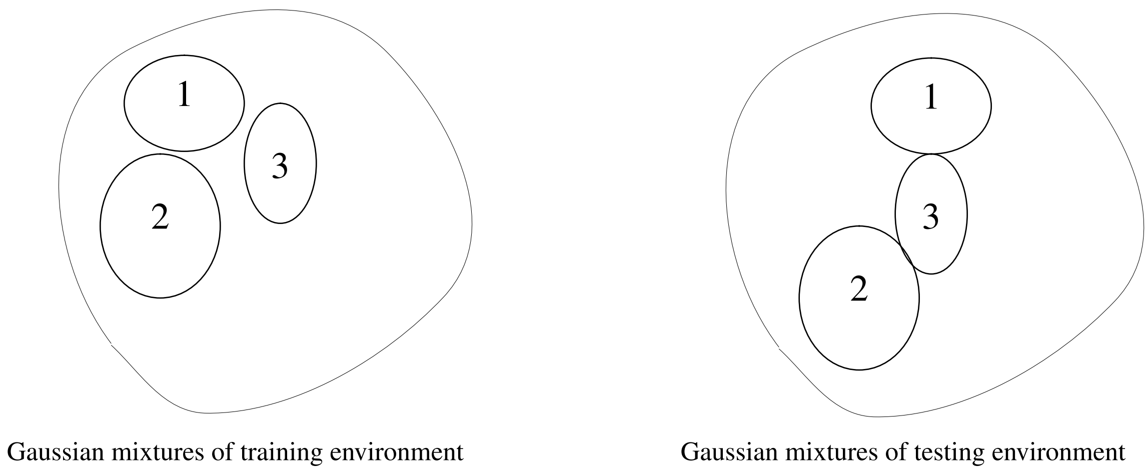


Figure 7-2. Illustration of codebook adaptation with updating only means while keeping variances unchanged.

Processing Algorithm	CMN	CMN+ MFCDCN	CMN+ DCCA2	CMN+ MFCDCN+ DCCA2
CLSTK (Training Mic)	7.6	7.6	7.9	7.6
Error Reduction	--	0.0	-3.9	0.0
Secondary-Mic Data	21.4	14.5	14.2	12.3
Error Reduction	--	32.2	33.6	42.5

Table 7-2. Results of DCAA2 on the ARPA WSJ0-si_evl5. In this case, only the means vectors are re-estimated to adapt to the target testing microphones.

Table 7-2 compares the performance of DCCA2 updating the means only while keeping variances unchanged. It is shown in Table 7-2 that DCCA2 achieves a 33.6% reduction of error rate from 21.4% to 14.2% when only the mean vectors of the secondary-microphone data are re-estimated. When we combine DCCA2 with MFCDCN, a complementary improvement is obtained with error rate reduced to 12.3%.

The results in Table 7-2 agree with our belief that mean vectors in Gaussian mixtures are more susceptible to changes in environments than variances, as only updating the mean vectors in DCCA2 can result in a substantial improvement. The table also shows that environmental variability may have a more obvious impact on the shift than on the scaling of each cluster in acoustic space. A comparison of Table 7-2 to Table 7-1 shows that DCCA2 achieves similar but slightly better result than DCCA.

In general, re-estimation of variances as well as mean vectors might be helpful in characterizing the possible shift and scaling of each vector codeword in the acoustic space. On the other hand, updating mean vectors can be more beneficial in the situation where only a limited amount of data is available to transform the codebook. Therefore, we speculate that the better results obtained above using fixed variances are partly attributed to the relatively poor estimation of variances due to the lack of enough adaptation training data for the various alternate microphones.

In summary, our experiments reveal that DCCA and DCCA2 can be used to address the issue of environmental robustness as they produce comparable performance to that of MFCDCN. Furthermore, greater benefit can be obtained when they are applied in conjunction with MFCDCN. With mean vectors and variances to be updated, DCCA is similar to the approach of BBN's approach, called tied mixture normalization (TMN) [96].

In the next section, we will examine the use of dual-channel codebook adaptation in the situations where the target testing environment does not exist in the set of prototype environments.

7.2.3. Dual-Channel Codebook Adaptation For Unseen Environments

Our previous experiments showed that the performance of a compensation algorithm degrades if the target testing environment does not resemble any prototype environment in the training set. Although Table 7-2 shows that dual-channel codebook adaptation produces comparable results to that of MFCDCN and PDCN, we are curious about whether improvement from dual-channel codebook adaptation is obtained when unseen environments are encountered.

To this end, all testing environments in the test data were excluded from the set of prototype environments, so that no data from the actual testing microphones were to derive the transformations. The environment is selected using the Gaussian environment classifier described in Section 5.3.3., and the corresponding adapted codebook is used in the recognition process. In this case, the chosen codebook is from an incorrect environment but it most resembles the acoustical properties of incoming testing data in the sense of maximum likelihood among all prototype environments.

Processing Algorithm	CMN	CMN+ MFCDCN	CMN+ IMFCDCN	CMN+ DCCA2	CMN+ MFCDCN +DCCA2	CMN+ IMFCDCN +DCCA2
CLSTK (Training Mic)	7.6	7.6	7.8	7.6	7.6	7.6
Error Reduction	--	0.0	-2.6	0.0	0.0	0.0
Secondary-Mic data	21.4	16.1	15.6	15.3	15.8	15.0
Error Reduction	--	24.8	27.1	28.5	26.2	29.9

Table 7-3. Result for DCCA2 in different combinations with MFCDCN and IMFCDCN for the ARPA WSJ0-si_evl5 test data. Note that all testing microphones are excluded from the set of prototype environments.

Table 7-3 compares the results of DCCA2 in different combinations of MFCDCN and IMFCDCN when all testing microphones are excluded from the set of prototype environments. It shows that DCCA2 can reduce the error rate from 21.4% to 15.3% for the system with CMN, a 28.5% error reduction. It indicates that DCCA2 enables the system to obtain a greater degree of environmental robustness by using a codebook from some other environment that better characterizes the acoustical space than by using the original codebook.

Table 7-3 shows that no significant improvement is obtained by combining DCCA2 with IMFCDCN when the testing environments are excluded from the prototype data. This may be due to the reduction of environmental variability provided by IMFCDCN.

7.3. Baum-Welch Codebook Adaptation

In this section, we propose and evaluate a second approach to implementing codebook adaptation. This approach is based on the Baum-Welch algorithm [7,59] that employs the contextual contents of the adaptation utterances, and is referred to as the Baum-Welch Codebook Adaptation (BWCA) algorithm. Compared to DCCA, it has the advantage that adaptation can be performed without the use of a stereo-recorded database

7.3.1. Baum-Welch Estimation

For speech recognition systems based on the HMM approach, an iterative procedure known as the Baum-Welch algorithm or the forward-backward algorithm [7,59] is used for learning. In the framework of SPHINX-II [40, 42], mean vectors and covariances, along with senones [41], are re-estimated and updated using the Baum-Welch algorithm in each iteration of training process as follows,

$$\bar{\mu}_k = \frac{\sum_t \sum_i \zeta_t(i,k)\, \boldsymbol{x}_t}{\sum_t \sum_i \sum_k \zeta_t(i,k)} \tag{7.8}$$

$$\bar{\Sigma}_k = \frac{\sum_t \sum_i \zeta_t(i,k)\, (\boldsymbol{x}_t - \bar{\mu}_k)\, (\boldsymbol{x}_t - \bar{\mu}_k)^T}{\sum_t \sum_i \sum_k \zeta_t(i,k)} \tag{7.9}$$

where $\zeta_t(i,k)$ is the probability that at time t, mixture k is chosen with the transition i. More details about the Baum-Welch algorithm can be found in [38, 42].

To compensate for changes in acoustical environments, Equation (7.8) and Equation (7.9) can be used to transform the mean vectors and covariances iteratively to better characterize the feature space of the adaptation utterances. From the viewpoint of system training, it would be desirable to train all model parameters using an ensemble of adaptation data from every prototype testing environments. This requires a large amount of data for a reliable estimate of all parameters including codebooks and output probabilities, which is typically unobtainable for prototype testing environments.

As an expediency, we can re-estimate the codebook while still keeping senones unchanged. Figure 7-3 shows the block diagram of Baum-Welch Codebook Adaptation (BWCA). The training procedure of BWCA is described in Figure 7-4.

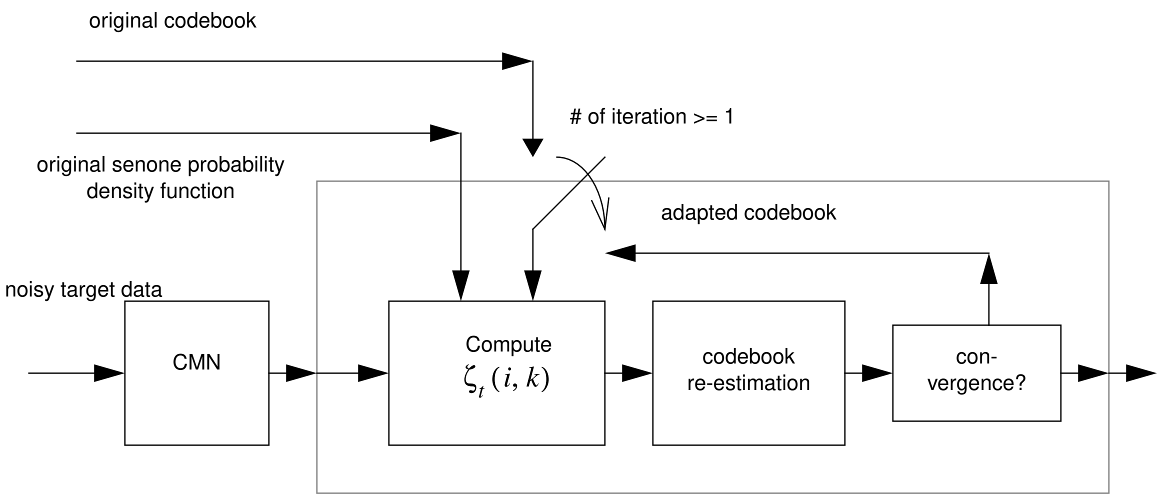


Figure 7-3. Block diagram of BWCA. Dashed block stands for step 2 described in Figure 7-4.

In this dissertation, we perform BWCA by adapting the mean vectors while keeping variances unchanged, based on our experience from DCCA. For SPHINX-II with 7000 senones, the number of parameters to be re-estimated in BWCA is reduced to 13,312 floating-point numbers (4 features x 256 vectors/feature x 13 floats/vector). In contrast, the normal training procedure needs to estimate 7,168,000 floating-point numbers (4 features x 7000 senones/feature x 256 floats/senone) for the senonic parameters plus 13,312 floats for means and variances.

In general, BWCA can be regarded as a simplification to the normal training process, and it differs from the normal training process in the following ways: (1) Only a small fraction of system parameters, mean vectors in this case, are to be updated from the initial models, (2) Only a small amount of adaption utterances is required for codebook adaptation as opposed to the training procedure.

1. Locate adaptation utterances with transcription and model parameters of system trained on clean training data to be used for each of prototype environments.

2. For each testing environment e,

   (a) Initialize model parameters by using the original system.

   (b) Compute $\zeta_t(i, k)$, the probability that at time t, mixture k is chosen with the transition i using model parameters of the original system.

   (c) Re-estimate the environment-specific mean vectors and/or co-variances using Equation (7.8).

   (d) Replace the mean vectors with the newly re-estimated means vectors from step (c).

   (e) Go to step (b) until the convergence criterion or desired number of iteration is met.

Figure 7-4. The training procedure of BWCA.

Processing Algorithm	CMN	CMN+ MFCDCN	CMN+ BWCA	CMN+ MFCDCN+ BWCA
CLSTK (Training Mic)	7.6	7.6	7.9	7.6
Error Reduction	--	0.0	-3.9	0.0
Secondary-Mic Data	21.4	14.5	16.7	14.1
Error Reduction	--	32.2	22.0	34.1

Table 7-4. Comparison of Baum-Welch codebook adaptation (BWCA) to MFCDCN for the ARPA WSJ0-si_evl5 test data. The number of iterations used for re-estimation is 4 in this table.

Table 7-4 compares results of BWCA and MFCDCN on the ARPA WSJ0-si_evl5 task. The number of iterations performed in BWCA is 4. It shows that BWCA produces a 22.0% error reduction by decreasing the error rate from 21.4% to 16.7%. When BWCA is applied in combination with MFCDCN to compensate for changes in environment, a 34.1% error reduction can be obtained relative to CMN with the error rate reduced to 14.1%. However, we note that this result may benefit most from the application of MFCDCN compensation vectors which are already well-estimated.

7.3.2. Baum-Welch Codebook Adaptation For Unseen Environments

As before, we are interested in the performance of BWCA in applications where the testing environment was not included in the set of prototype environments.

Processing Algorithm	**CMN**	**CMN+ MFCDCN**	**CMN+ IMFCDCN**	**CMN+ BWCA**	**CMN+ MFCDCN +BWCA**	**CMN+ IMFCDCN +BWCA**
CLSTK (training mic)	7.6	7.6	7.6	7.6	7.6	7.6
Error Reduction	--	0.0	0.0	0.0	0.0	0.0
Secondary-Mic	21.4	16.3	15.6	16.9	15.5	14.6
Error Reduction	--	23.8	27.1	21.0	27.8	31.8

Table 7-5. Results of BWCA in different combination. The same as Table 7-4 except that the testing environments are excluded from the corpus used to develop the compensation vectors.

Table 7-5 compares the results of BWCA in different combination with MFCDCN and IMFCDCN with the target testing environment excluded from the set of prototype testing environments. The procedure used is similar to that described in Section 7.2.3., with the testing microphones in the ARPA WSJ0-si_evl5 task excluded from the list of prototype environments. It is seen in Table 7-5 that BWCA reduces the error rate from 21.4% to 16.9% using the system with

CMN, a 21.0% error reduction. This indicates that BWCA can provide a high degree of environmental robustness by using a codebook from another acoustically similar environment.

7.4. Summary

In this chapter, we studied the technique of codebook adaptation for the problem of environmental variability. Two different approaches were proposed. The first approach is dual-channel codebook adaptation, which addresses the problem of mismatches between training and testing environment by deriving a new codebook for the testing environment using stereo-recording data. The second approach is the Baum-Welch codebook adaptation algorithm, which adapts the codebook to match the testing environment using adaptation utterances with transcriptions from the testing environment.

It can be shown that the approach of codebook adaptation is similar in concept to signal-enhancing compensation techniques. The techniques of codebook adaptation such as DCCA and BWCA differ from signal-enhancing algorithms such as BSDCN and MFCDCN in that the transformation between mismatched environments is characterized in terms of mean vectors and/or variances of the internal representation of the templates in the recognition system, instead of in the form of compensation vectors that are applied to the features that are input to the system.

We found that both DCCA and BWCA are effective in compensating for the changes in environment even when the testing environment does not exist in the set of prototype environments. It is also shown that both approaches can achieve comparable results to that of MFCDCN and PDCN. The result is an error rate of 14.2% rate for the dual-channel codebook adaptation and an error rate of 16.7% for the Baum-Welch codebook adaptation.

Complementary improvement can be obtained for codebook adaptation when it is used in conjunction with MFCDCN. The word error rate can be reduced from 14.2% without MFCDCN to 12.3% with MFCDCN for dual-channel codebook adaptation. Similarly, the error rate is reduced from 16.7% to 14.1 for the Baum-Welch codebook adaptation with MFCDCN.

Chapter 8
Summary and Conclusions

This dissertation presents our efforts to address the issue of environmental robustness in current speech recognition technology. Many lessons have been learned about the nature of environmental variabilities and several algorithms have been proposed to alleviate performance degradation. In this chapter, we summarize our observations and findings based on our experimental experiences with these techniques, review the major contributions of this work, and present several suggestions for future work.

8.1. Summary of Results

We performed a series of baseline experiments to probe the degree of performance degradation observed when changes in environments were encountered. The error rate for the state-of-the-art SPHINX-II system increased by four to five times without cepstral mean normalization when the microphone was switched from the standard close-talking training microphone to one of several alternate microphones. Even with cepstral mean normalization, the word error rates observed using the alternate microphone were three times larger than those observed with the close-talking training microphone. This level of degradation demonstrates a need for environment independence in speech recognition systems and, therefore, motivates our study of environmental robustness.

In order to examine environmental mismatches, experiments were carried out using the ARPA World-Street-Journal-based corpora, which presently serve as a common database for various research aspects within the ARPA community. In particular, the environmental variabilities in this task were characterized by changes of the recording microphone. We have found that the two primary sources of degradation encountered in this database are distortions due to the effects of additive noise and linear filtering.

Our experiments showed that robust speech recognition with respect to acoustical environments could be achieved by reducing mismatches between the acoustical space of the training environment and various testing environments.

We now summarize our major findings obtained using the compensation techniques described in this dissertation.

- Environmental independence can be achieved by using signal processing to adjust the features that characterize the incoming speech signal, (such as the BSDCN and MFCDCN algorithms), or by "adjusting" the internal parameters of the recognition system used during the search process (such as the IPDCN, DCCA, and BWCA algorithms).

- A major disadvantage of environment-dependent algorithms is the need for re-calibration for each new environment. This re-calibration requires *a priori* knowledge of the testing environment identity and the availability of simultaneously-recorded utterances for the particular environment. These two constraints can be alleviated by employing stereo data from a number of different environments in existing corpora. Thus, all of the environments in the training corpus can be utilized as prototype testing environments to provide information about the effects of acoustical variability on speech in unknown testing environments.

- The use of high-pass filtering of cepstral components helps to reduce the slow-varying spectral changes due to channel effects. In particular, cepstral mean normalization produced a 44% reduction in word error rate for speech from mismatched environments and a 6% reduction of errors for data from the training environment. CMN proves to be a simple and efficient way to increase the recognition accuracy in both clean and noisy conditions.

- BSDCN is a simple but effective technique that utilizes dynamic programming to derive an environment mapping based on correspondence of SNRs, eliminating the need for stereo-data. For systems with CMN, BSDCN can produce a 40% reduction of errors in SPHINX (D-HMM) and approximately a 12~15% error reduction in SPHINX-II (SC-HMM) for different tasks.

- The success of MFCDCN underscores the significance of knowledge that can be learned from prototype testing environments. Moreover, MFCDCN shows that a properly detailed characterization of environmental variability is more beneficial in compensation for unknown linear filtering and additive noise. For SPHINX-II including CMN, MFCDCN can generate a 32% error reduction. The improvement of IMFCDCN obtained in acoustically dissimilar testing environments from basis environments indicates that environment interpolation points in the right direction. For a task in which all testing environments are not covered by basis environments, IMFCDCN can produce a 27% error reduction compared to a 23% error reduction for MFCDCN.

- The use of phone-dependencies for characterizing environmental variability is promising. PDCN uses the discriminating ability of the HMM during the search process to produce an improvement of 21% in error reduction. Furthermore, PDCN presents a compensation paradigm which can be applied in combination with MFCDCN to produce produces a 40% error reduction relative to the baseline system with CMN. On the other hand, we note that no significant improvement is produced when PDCN correction vectors are tabulated separately as a function of SNR. This may be due to a high correlation between phonetic features and instantaneous SNR values.

- By adjusting mean vectors, codebook adaptation provides comparable results to MFCDCN. DCCA produces a 33% error reduction relative to the baseline system with CMN and BWCA reduces errors by 22%. Similarly, further improvements can be obtained by applying these codebook adaptation techniques in combination with MFCDCN. We also found that slightly better results are obtained with only the mean vectors are updated, compared to results obtained updating both mean vectors and covariance matrices.

8.2. Contributions

We summarize below the major contributions of this thesis.

- We proposed a series of compensation algorithms based on cepstral comparisons that can be used to achieve environmental robustness in an unsupervised fashion during the recognition phase. Our algorithms do not assume *a priori* knowledge about the identity of the testing environment. These algorithms can be applied to various environments whether the testing condition is matched to the training condition or not. We showed that adaptation to mismatched environments is achievable without going through a re-training process. Using a regular system trained for uncompensated clean speech, we demonstrated that our algorithms can maintain high accuracy obtained in a matched condition while improving recognition accuracy for unknown and mismatched conditions.

- We showed that BSDCN is effective in compensating for environmental variabilities by depending on the instantaneous frame SNR values when the correspondence of SNR values between environments is established using a dynamic programing technique. Compensation vectors can be obtained using a few adaptation utterances from the testing environment.

- We developed techniques such as MFCDCN and IFCDCN that exhibit substantial environmental robustness by use of detailed characterization of environmental differences between environments in combination with appropriate procedures for microphone selection or interpolation.

- We demonstrated that interpolating compensation vectors across acoustical environments can improve performance when the testing environment is not included in the ensemble of environments used to develop the compensation vectors.

- We demonstrated the feasibility of characterizing environmental differences in terms of phonetic labels using PDCN, which enables us to utilize the discrimination ability of the recognition system for environmental compensation.

- We studied and evaluated the technique of codebook adaptation in the context of environmental adaptation. It was shown that codebook adaptation can achieve comparable results to other signal-processing techniques that combat acoustical mismatches by adapting the speech features to the training environment.

- We introduced a compensation paradigm combining signal-processing compensation techniques with search-based compensation techniques. Our results showed that this is promising since further error reduction can be achieved by combining these two kinds of approaches. Although these two kinds of approaches are considered to accomplish the same task they can be complementary to each other for completeness of compensation.

8.3. Suggestions for Future Work

While our experiments have achieved a great deal of improvement in the domain of microphone independence, the results still demonstrate the need for several areas of further study.

- In the database used in this dissertation, the additive noise present is primarily ambient noise. Although the energy levels of ambient noise output by different microphones may be very different, the database is does not provide a broad coverage of different noises. For example, non-stationary noise is not encountered and the energy level of background noise is not tuned to different SNR levels. It may be valuable to evaluate algorithms on another corpus that characterizes different types of noises. Another alternative would be to use a database created by contaminating clean speech using various types of artificially-added noises at different SNR levels.

- Our experiments showed that MFCDCN outperformed BSDCN in all testing conditions partly because of the use of an additional partition dimension, VQ label, for more detailed characterization of environmental mismatches. Inclusion of another partition dimension may be helpful if an appropriate attribute is used. Though it is questionable whether the gender of speakers has a direct impact on characterization of environmental variability, the use of gender information should represent a topic to study in future research.

- The compensation procedure in our algorithms depends only on the current frame. However, information in adjacent frames is believed to be helpful in capturing fundamental variabilities because it is observed that temporal transitional relationships between adjacent frames are important in speech recognition. Performance may be further improved by exploiting information on the temporal evolution of cepstral coefficients in developing compensation vectors.

- In this dissertation, we did not specifically address the issue of the amount of adaptation data for each algorithm for every environment. The issue of a how much adaptation data is needed should be further investigated in future work.

- Other approaches for environmental clustering should be explored in future research. In the current implementation of our algorithms, all existing microphones in the CSR corpus are processed separately and a set of compensation vectors is produced for each microphone respectively. While this helps maintain distinctive compensation vectors as long as enough data are available, some microphones might have very similar characteristics in frequency response and noise-canceling ability. In these cases, clustering similar microphone into a microphone class could improve the quality of compensation due to the availability of more adaption data for each class. One potential approach is to represent each environment parametrically, as in an HMM or VQ codebook, and perform clustering on these representations.

- One of the potential problems with our algorithms is the dependence on stereo data from already-available corpora, such as the CSR task, to constitute prototype environments used to develop compensation vectors. Other techniques should be developed to cope with situations in which sources of degradation in the testing environment are very different from what is available in any of the development environments.

- Currently, PDCN is implemented in a way such that only the static features related to the current frame are normalized. Because it is difficult to consider phonetic information across time frames, we opt to normalize cepstrum only in PDCN. It may well be the case, however, that normalization of dynamic features such as differenced cepstrum and second-order differenced cepstrum is helpful in providing further improvement. A possible extension of PDCN that can compensate across different frames and deal with co-articulatory effects should be studied in the future.

- The use of presumed phonetic identity was plausible based on the results of PDCN. One alternative would be to implement a "state-dependent" or "senone-dependent" normalization technique.

- Unlike MFCDCN, PDCN did not yield a substantial improvement by including the instantaneous frame SNRs. This can be due to the possibly high correlation between phonetic identities and frame SNRs. It is possible that a complimentary improvement can be still achieved by accordingly adjusting the dynamic range of SNRs in PDCN and DCCA.

- Both IMFCDCN and IPDCN are based on a Gaussian distribution for the interpolation of basis environments. Since this may be an invalid assumption, it would be desirable to investigate other approaches to appropriately interpolate basis environments.

- This dissertation copes with environmental adaptation for microphone independence. There are other areas that present more challenging issues for acoustical robustness. They include speech recognition over band-limited telephone lines, in a moving automobile with background interference, and in a very noisy factory with significant impulsive noises. More standard corpora containing various types of environment variabilities should be explored and evaluated for a more thorough study of environmental robustness.

Appendix A
Phone Table Used By SPHINX-II For WSJ

In this appendix, we tabulate the phone labels for all 63 basic phones used in the current SPHINX-II system. There are 50 lexical phones, 3 silence models, and 10 noise models for non-speech voices. Examples are also given in this table.

Phone	Example	Phone	Example	Phone	Example
/+BUMP+/	*door bump*	/+CLICK+/	*tongue click*	/+EXHALE+/	*exhalation*
/+INHALE+/	*inhalation*	/+NOISE+/	*office noise*	/+POP+/	*pop noises*
/+RUSTLE+/	*paper rustle*	/+SMACK+/	*tongue smack*	/+SWALLOW+/	*saliva swallow*
/+UH+/	*uh, ah, em, etc.*				
/SIL/	*(middle silence)*	/SILb/	*(begin sil)*	/SILe/	*(end sil)*
/IY/	*b*__ea__*t*	/R/	__r__*ed*	/K/	__k__*ick*
/IH/	*b*__i__*t*	/Y/	__y__*et*	/BD/	*ro*__b__
/EH/	*b*__e__*t*	/W/	__w__*et*	/DD/	*ba*__d__
/AE/	*b*__a__*t*	/ER/	*b*__ir__*d*	/GD/	*do*__g__
/IX/	*ros*__es__	/AXR/	*din*__er__	/PD/	*wra*__p__
/AX/	*th*__e__	/M/	__m__*om*	/TD/	*si*__t__
/AH/	*b*__u__*t*	/N/	__n__*on*	/KD/	*sic*__k__
/UW/	*b*__oo__*t*	/NG/	*si*__ng__	/TS/	*ca*__ts__
/UH/	*b*__oo__*k*	/CH/	__ch__*urch*	/Z/	__z__*oo*
/AO/	*b*__ou__*ght*	/JH/	__j__*udge*	/ZH/	*mea*__s__*ure*
/AA/	*c*__o__*t*	/DH/	__th__*ey*	/V/	__v__*ery*
/EY/	*b*__ai__*t*	/B/	__b__*ob*	/F/	*brie*__f__
/AY/	*b*__i__*te*	/D/	__d__*ad*	/TH/	__th__*ief*
/OY/	*b*__oy__	/DX/	*bu*__tt__*er*	/S/	__s__*ix*
/AW/	*ab*__ou__*t*	/G/	__g__*ag*	/SH/	__sh__*oe*
/OU/	*b*__oa__*t*	/P/	__p__*op*	/HH/	__h__*ay*
/L/	__l__*ed*	/T/	__t__*ot*		

Table A-1.The basic phone set used in the SPHINX-II system.

Appendix B
Statistical Significance Test

The NIST "standard" benchmark scoring program [52] is employed in this appendix to compare several major results obtained using various algorithms proposed in this thesis. Table B-1 summarizes the comparisons of performance in terms of statistical significance on the ARPA WSJ0-si_evl5 task. These comparisons were made using the matched-pairs test due to the assumption of error independence [28, 52] for continuous speech recognition.

System ID	Processing Algorithm	Corresponding Word Error Rate (%)
sys1	CMN (baseline)	21.4
sys2	CMN+MFCDCN	14.5
sys3	CMN+IFCDCN	15.0
sys4	CMN+PDCN	16.9
sys5	CMN+MFCDCN+PDCN	12.9
sys6	CMN+DCCA	14.2
sys7	CMN+MFCDCN+DCCA	12.3

	sys1	sys2	sys3	sys4	sys5	sys6	sys7
sys1	--	sys2	sys3	sys4	sys5	sys6	sys7
sys2	--	--	same	sys2	sys5	same	sys7
sys3	--	--	--	sys3	sys5	same	sys7
sys4	--	--	--	--	sys5	sys6	sys7
sys5	--	--	--	--	--	sys5	same
sys6	--	--	--	--	--	--	sys7
sys7	--	--	--	--	--	--	--

Table B-1. Comparison matrix showing the results of the matched-pairs test. If the test results indicate that the difference is significant, the identity of the "better" system is in the corresponding box. If the difference in performance is not significant, "same" is used.

Appendix C
Confusion Matrix for Environment Selection

This appendix provides detailed comparison between the two environment selection procedures used in this thesis, the selection-by-compensation procedure and the Gaussian environment classifier. Table C-1 lists the indices for all prototype environments used in this appendix. Details about the prototype environments can be also found in Table 3-1. The ARPA WSJ0-si_evl5 task is used, in which the Sennheiser close-talking microphone (m1) is used as the primary channel to collect 330 utterances. Three other microphones are used to collect the secondary-microphone data, including AT&T 720 (m1), RadioShack HighBall (m3), and Shure SM91 (m4).

Environment ID	Recording microphone	Environment ID	Recording microphone
m1	Sennheiser HMD-410(414)	m9	Nakamichi CM100
m2	AT&T 720	m10	Panasonic KXT2365
m3	RadioShack Highball	m11	RadioShack Omni
m4	Shure SM91	m12	R.S. 33-1063 Tie-Pin
m5	AKG D541	m13	R.S. 33-1052 tie-clip
m6	AT&T 5400	m14	Sony ECM155
m7	Crown PCC-160	m15	Sony ECM-50PS
m8	Crown PZM-6FS	m16	Sony ECM-55

Table C-1. Environment identities for the ARPA WSJ0 task.

Table C-2 shows the confusion matrix of environment selection for the selection-by-compensation procedure using the ARPA WSJ0-si_evl5 task including the data recorded using secondary microphones and Sennheiser close-talking microphone, and Table C-3 shows the confusion matrix for the Gaussian environment classifier on the same task.

output microphone		m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11 ~ m16	Total
input microphone	m1	320	--	7	--	--	--	--	1	2	--	--	330
	m2	--	107	--	--	--	1	--	--	--	17	--	125
	m3	--	--	82	--	--	--	--	--	--	--	--	82
	m4	--	--	15	35	--	--	50	--	23	--	--	123

Table C-2. Confusion matrix for the *selection-by-compensation* procedure on the WSJ0-si_evl5 task. The procedure of selection-by-selection is applied to both noisy speech recording using secondary microphones and clean speech recorded using the Sennheiser close-talking microphone. The environment identities, m1,...m16, are defined in Table C-1.

output mic		m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11 ~ m16	Total
input microphone	m1	291	--	31	8	--	--	--	--	--	--	--	330
	m2	--	91	--	--	--	--	--	--	--	34	--	125
	m3	1	--	81	--	--	--	--	--	--	--	--	82
	m4	--	--	--	123	--	--	--	--	--	--	--	123

Table C-3. Confusion matrix for the *Gaussian environment classifier* method on the WSJ0-si_evl5 task. The method of Gaussian environment classifier is applied to both noisy speech recording using secondary microphones and clean speech recorded using the Sennheiser close-talking microphone.

Appendix D
Breakdown Of Results By Microphones

In this appendix, we tabulate the microphone-by-microphone breakdown of the major recognition results in this thesis using different combinations of compensation algorithms and the secondary-microphone testing data. Table D-1 compares results for the ARPA WSJ0-si_evl5 task. Three secondary microphones were used to collect 330 testing utterances, including the AT&T 720 (ATT 720), RadioShack HighBall (RSHB), and Shure SM91 (SM91) microphones. Details about these microphones are listed in Table 3-2 in Chapter 3.

Processing Algorithm	ATT 720	SM91	RSHB	Overall of 2nd-mic
CMN	38.6	13.1	8.5	21.4
CMN+MFCDCN	24.5	10.8	5.7	14.6
CMN+IFCDCN	24.7	10.6	7.3	15.0
CMN+PDCN	25.2	15.5	7.0	16.9
CMN+MFCDCN+PDCN	19.6	11.0	5.8	12.9
CMN+DCCA2	22.1	11.4	6.7	14.2
CMN+MFCDCN+DCCA2	18.2	10.7	6.3	12.3
CMN+BWCA	27.0	13.5	6.6	16.7
CMN+MFCDCN+DCCA	22.7	11.0	6.2	14.1

Table D-1. Detailed microphone-by-microphone breakdown of results (word error rates) with CMN on the ARPA WSJ0-si_evl5 task.

Table D-2 compares results for the ARPA WSJ1-si_dt_s5 task. Nine secondary microphones were used to collect 216 testing utterances, including the AT&T 712 (ATT 712), AT&T 720 (ATT 720), Audio-Technica 853a (AT 853a), Radioshack 33-992D (RS 33992D), Radioshack Pro (RS Pro), SGI clip-on (SGI), Shure WL84 (WL84), Sony ECM-K7 (ECM-K7), and Sun monitor (Sun) microphones. Details about these environments are listed in Table 3-3 in Chapter 3.

Processing Algorithm	CMN	CMN+ BSDCN	CMN+ MFCDCN	CMN+ IMFCDCN	CMN+ MFCDCN +PDCN	CMN+ MFCDCN +DCCA2
ATT 712	23.2	19.3	11.0	8.9	12.2	10.4
ATT 720	61.7	53.1	41.5	43.9	38.3	28.2
AT853a	13.6	11.5	13.1	13.3	12.9	11.9
RS 33992D	13.5	14.3	11.7	11.4	11.9	13.8
RS Pro	14.4	14.4	14.4	12.4	15.2	13.2
SGI	26.8	26.5	21.8	19.8	24.2	25.5
WL84	13.1	11.9	11.7	10.8	11.4	13.1
ECM-K7	28.5	22.8	22.4	24.2	24.6	25.3
Sun	25.4	17.1	19.2	17.4	18.3	19.5
Overall (2nd-mic)	23.0	20.0	17.8	17.2	17.8	17.0

Table D-2. Detailed microphone-by-microphone breakdown of results (word error rates) with CMN on the ARPA WSJ1-si_dt_s5 task.

REFERENCES

[1] A. Acero, "Acoustical and Environmental Robustness in Automatic Speech Recognition", Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Sept. 1990.

[2] A. Acero, and R. Stern, "Robust Speech Recognition by Normalization of the Acoustic Space", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 893-896, May, 1991.

[3] F. Alleva, X. Huang, and M. Hwang, "An Improved Search Algorithm for Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II 307-310, May, 1993.

[4] L. Bahl, F. Jelinek, and R. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179-190, March 1983.

[5] J. Baker, "Stochastic Modeling as a Means of Automatic Speech Recognition", Ph.D. Thesis, Computer Science Department, Carnegie Mellon University, April 1975.

[6] R. Bakis, "Continuous Speech Recognition via Centisecond Acoustic States", *91st Meeting of the Acoustical Society of America*, April, 1976.

[7] L. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes", *Inequalities* 3:1-8, 1972.

[8] V. Beattie, and S. Young, "Hidden Markov Model State-Based Noise Cancellation", Technical Report, Engineering Department, Cambridge University, Feb. 1992.

[9] J. Bellegarda, P. de Souza, A. Nadas, D. Nahamoo, M. Picheny, and L. Bahl, "Robust Speaker Adaptation Using a Piecewise Linear Acoustic Mapping", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-445 ~ I-448, March, 1992.

[10] S. Boll, "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-27, No. 2, pp. 113-120, April 1979

[11] M. Carey, H. Chen, A. Descloux, J. Ingle, and K. Park, "End Office Connection Study: Analog Voice and Voiceband Data Transmission Performance Characterization of the Public Switched Network", *The Bell System Technical Journal*, 63, Nov. 1984.

[12] B. Carlson, "A Projection-Based Measure for Automatic Speech Recognition in Noise", Ph.D. Thesis, Georgia Institute of Technology, Nov. 1991.

[13] B. Chigier, "Phonetic Classification on Wide-Band and Telephone Quality Speech", *Proceedings of DARPA Speech and Natural Language Workshop*, pp. 291-295, Feb., 1992.

[14] Y. Chow, M. Dunham, O. Kimball, M. Krasner, F. Kubala, J. Markoul, S. Roucos, and R. Schwartz, "BYBLOS: The BBN Continuous Speech Recognition System", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp.89-92, April, 1987.

[15] N. Dal Degan, and C. Prati, "Acoustic Noise Analysis and Speech Enhancement Techniques for Mobile Radio Applications", S*ignal Processing*, 15, pp43-56, 1988.

[16] S. Das, R. Bakis, A. Nadas, D. Nahamoo, and M. Picheny, "Influence of Background Noise and Mi-

crophone on the Performance of the IBM Tangora Speech Recognition System", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 71-74, April, 1993.

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

[18] Y. Ephraim, J. Wilpon, and L. Rabiner, "A Linear Predictive Front-End Processor for Speech Recognition in Noisy Environments", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1324-1327, April, 1987.

[19] M. Feng, "Fast Speaker-Adaptive Training For Large-Vocabulary Speech Recognition", Ph.D. Thesis, Northeastern University, June 1989.

[20] M. Feng, R. Schwartz, F. Kubala, and J. Makhoul, "Iterative Normalization for Speaker Adaptive Training in Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Paper S12.4, May 1989.

[21] J. Flanagan, J. Johnston, R. Zahn, and G. Elko, "Computer-steered Microphone Arrays for Sound Transduction in Large Rooms", *the Journal of Acoustical Society of America*, Vol. 78, pp. 1508-1518, Nov. 1985.

[22] J. Flanagan, R. Mammone, and G. Elko, "Autodirective Microphone Systems For Natural Communication with Speech Recognizers", *Proceedings of DARPA Speech and Natural Language Workshop*, pp. 170 - 175, Feb. 1991.

[23] S. Furui, "Unsupervised Speaker Adaptation Based on Hierarchical Spectral Clustering", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 37, No. 12, pp. 1923-1930, Dec. 1989

[24] M. Gale, and S. Young, "An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-233 - I-236, March, 1992.

[25] J. Gauvain, L. Lamel, G. Adda, and M.Adda-Decker, "The LIMSI Continuous Speech Dictation System: Evaluated on the ARPA Wall Street Journal Task", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 557-560, April, 1994.

[26] W. Gaylor, *Telephone Voice Transmission. Standards and Measurements*, Prentice Hall Inc., 1989.

[27] O. Ghitza, "Auditory Nerve Representation as a Front-End for Speech Recognition in a Noisy Environment", *Computer Speech and Language*, Vol. 1, pp. 109-130, 1986.

[28] L. Gillick, and S. Cox, "Some Statistical Issues in the Comparison of Speech Recognition Algorithms", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 532-535, May 1989.

[29] R. Gray, "Vector Quantization", *IEEE ASSP Magazine* 1(2):4-29, April 1984.

[30] N. Hanai, "Speech Recognition in the Automobile", M.S. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, May 1993.

[31] Y. Haneda, S. Makino, and Y. Kaneda, "Modeling of a Room Transfer Function Using Common Acoustical Poles", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II 213-216, March, 1992

[32] J. Hansen, "Adaptive Source Generator Compensation and Enhancement for Speech Recognition

in Noisy Stressful Environment", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 71-74, April, 1993.

[33] B. Hanson, and H. Wakita, "Spectral Slope Distance Measures With Linear Prediction Analysis for Word Recognition in Noise", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. ASSP-35, pp. 968-973, July 1987.

[34] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP Speech Analysis Technique", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-121 - I-124, March, 1992.

[35] H. Hermansky, N. Morgan, and H. Hirsch, "Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. II-83 - 86, April, 1993.

[36] H. Hon, "Vocabulary-Independent Speech Recognition: the VOCIND System", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Feb. 1992.

[37] X. Huang, and M. Jack, "Semi-Continuous Hidden Markov Models with Maximum Likelihood Vector Quantization", *IEEE Workshop on Speech Recognition*, 1988.

[38] X. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh, U.K., 1990.

[39] X. Huang, "Speaker Normalization for Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-465 ~ I-468, March, 1992.

[40] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee, R. Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview", *Computer Speech and Language*, vol. 2, pp. 137-148, 1993.

[41] M. Hwang, and X. Huang, "Shared-Distribution Hidden Markov Models for Speech Recognition", *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 414-420, 1993.

[42] M. Hwang, "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Dec. 1993.

[43] M. Hwang, R. Rosenfeld, E. Thayer, R. Mosur, L. Chase, R. Weide, X. Huang, and F. Alleva, "Improving Speech-Recognition Performance via Phone-Dependent VQ Codebooks and Adaptive Language", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 549-552, April, 1994.

[44] F. Jelinek, "Continuous Speech Recognition by Statistical Methods", *Proceedings of the IEEE* 64(4):532-556, April 1976.

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

[46] B. Juang, and L. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33, pp. 1404-1413, 1985.

[47] D. Klatt, "A Digital Filter for Spectral Matching", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 573-576, 1976.

[48] F. Kubala, R. Schwartz, and C. Barry, "Speaker Adaptation From a Speaker-Independent Training Corpus", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 137-140, April, 1990.

[49] F. Kubala, and R. Schwartz, "Improved Speaker Adaptation Using Multiple Reference Speakers", *Proceedings of International Conference on Spoken Language Processing*, pp. 153-156, Nov. 1990.

[50] F. Kubala, A. Anastasakos, J. Makhoul, L. Nguyen, R. Schwartz, and G. Zavaliagkos, "Comparative Experiments on Large Vocabulary Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 561-564, April, 1994.

[51] N. Laird, A. Dempster, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM algorithm", *Annual Royal Statistics Society*, 1-38, Dec. 1987.

[52] I. Lecomte, M. Lever, J. Boudy, and A. Tassy, "Car Noise Processing for Speech Input", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 512-515, May, 1989.

[53] C. Lee, L. Rabiner, R. Pieraccini, and J. Wilpon, "Acoustic Modeling for Large Vocabulary Speech Recognition" *Computer Speech and Language*, vol. 4, 1990.

[54] C. Lee, C. Lin, and B. Juang, "A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models", *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 39, No. 4, pp. 806-814, April, 1991

[55] K. Lee and H. Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 123-126, April 1988.

[56] K. Lee, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, April. 1988.

[57] K. Lee, H. Hon, and R. Reddy, "An Overview of the SPHINX Speech Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 35-45, Jan. 1990.

[58] S. Lerner and B. Mazor, "Telephone Channel Normalization for Automatic Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-261 ~ I-264, March, 1992

[59] S. Levinson, L. Rabiner, M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Function on a Markov Process to Automatic Speech Recognition", *the Bell System Technical Journal* 62(4), April, 1983.

[60] J. Lim, "edited", *Speech Enhancement*, Prentice-Hall Inc., 1983

[61] Y. Linde, A. Buzo, R. Gray, "An Algorithm for Vector Quantization Design", *IEEE Transactions on Communication* COM-28(1): 84-95, Jan. 1980.

[62] R. Lippmann, E. Martin, and D. Paul, "Multi-Style Training for Robust Isolated-Word Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 705-708, April 1987.

[63] F. Liu, A. Acero, and R. Stern, "Efficient Joint Compensation of Speech For the Effects of Additive Noise and Linear Filtering", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-257 - I-260, March, 1992

[64] F. Liu, R. Stern, X. Huang, and A. Acero, "Efficient Cepstral Normalization for Robust Speech Recognition", *Proceedings of ARPA Speech and Natural Language Workshop*, pp. 69 - 74, March, 1993.

[65] F. Liu, P. Moreno, R. Stern, and A. Acero, "Signal Processing for Robust Speech Recognition", *Proceedings of ARPA Human Language Technology Workshop*, March, 1994.

[66] F. Liu, R. Stern, A. Acero, and P. Moreno, “Environment Normalization for Robust Speech Recognition using Direct Cepstral Comparison”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 85-88, April, 1994.

[67] B. Lowerre, and D. Reddy, *The Harpy Speech Understanding System*, Prentice-Hall Inc., 1980.

[68] R. Lyon, “Speech Recognition in Scale Space”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1265-1268, April, 1987.

[69] D. Mansour and B. Juang, “The Short-Time Modified Coherence Representation and Noisy Speech Recognition”, *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 37, pp. 795-804, June 1989.

[70] D. Mansour and B. Juang, “A Family of Distortion Measures Based Upon Projection Operation For Robust Speech Recognition”, *IEEE Transactions on Acoustics, Speech, and Speech, and Signal Processing, Signal Processing*, vol. 37, pp. 1659-1671, Nov. 1989.

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

[72] G. McLachlan, and K. Basford, *Mixture Models: Inference And Application to Clustering*, M. Dekker., 1988.

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

[74] L. Meumeyer, and M. Weintraub, “Probabilistic Optimum Filtering for Robust Speech Recognition”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 417-420, March, 1994.

[75] P. Moreno, *Personal Communications*, “unpublished”, 1993.

[76] P. Moreno, and R. Stern, “Sources of Degradation of Speech Recognition in Telephone Environments”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 109-112, April, 1994.

[77] H. Murveit, M. Butzberger, and M. Weintraub, “Reduced Channel Dependence for Speech Recognition”, *Proceedings of DARPA Speech and Natural Language Workshop*, pp. 280-284, Feb., 1992.

[78] A. Nadas, D. Nahamoo, and M. Picheny., “Adaptive Labeling: Normalization of Speech by Adaptive Transformation Based on Vector Quantization”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 521-524, April, 1988.

[79] B. Necioglu, M. Ostendorf, and J. Rohlicek, “A Bayesian Approach to Speaker Adaptation for the Stochastic Segment Model”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I-437 ~ I-440, March, 1992.

[80] N. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Co., 1980.

[81] Y. Ohshima, “Environmental Robustness in Speech Recognition using Physiologically-Motivated Signal Processing”, Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Dec. 1993.

[82] A. Oppenheim, E. Weinstein, K. Zangi, M. Feder, and D. Gauger, “Single Sensor Active Noise Cancellation Based on the EM Algorithm”, *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 277-280, March, 1992.

[83] D. Paul, and J. Baker, "The Design of the Wall Street Journal-based CSR Corpus", *Proceedings of ARPA Speech and Natural Language Workshop*, pp. 357-362, Feb., 1992.

[84] D. Pallett, W. Fisher, and J. Fiscus, "Tools for the Analysis of Benchmark Speech Recognition Tests", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 97-100, April, 1990.

[85] P. Peterson, "Adaptive Array Processing for Multiple Microphone Hearing Aids", Ph.D. Thesis, Massachusetts Institute of Technology, 1989.

[86] J. Picone, G. Doddington, and D. Pallett, "Phone-mediated Word Alignment for Speech Recognition Evaluation", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-38, pp. 559-562, March 1990.

[87] D. Pisoni, and R. Bernacki, H. Nusbaum, and M. Yuchtman, "Some Acoustic-Phonetic Correlates of Speech Produced in Noise", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1581-1584, March, 1985.

[88] J. Porter, and S. Boll, "Optimal Estimators for Spectral Restoration of Noisy Speech", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 18A.2.1-18A.2.4, March, 1984.

[89] L. Rabiner, and B. Juang, "An Introduction to Hidden Markov Models", *IEEE ASSP Magazine* 3(1):4-16, Jan. 1986.

[90] D. Rao, "Speech Recognition with a noise-adapting codebook", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1139 - 1142, April, 1987.

[91] G. Rigoll, "Speaker Adaptation for Large Vocabulary Speech Recognition Systems Using Speaker Markov Models", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5-8, May, 1989.

[92] D. Rtischev, "Speaker Adaptation in a Large Vocabulary Speech Recognition System", M.S. Thesis, M.I.T., January 1989.

[93] H. Sakoe, and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. ASSP-26, pp. 43-49, April 1978

[94] R. Schwartz, Y. Chow, S. Roucos, M. Krasner, J. Makhoul, "Improved Hidden Markov Modeling of Phonemes for Continuous Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1984

[95] R. Schwartz, and Y. Chow, "The Optimal N-Best Algorithm: An Efficient Procedure for Finding Multiple Sentence Hypotheses", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1990.

[96] R. Schwartz., T. Anastasakos, F. Kubala, J. Makhoul, L. Nguyen, and G. Zavaliagkos, "Comparative Experiments on Large Vocabulary Speech Recognition", *Proceedings of ARPA Human Language Technology Workshop*, March, 1993.

[97] S. Seneff, "A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing", *Journal of Phonetics,* Vol. 16, pp. 55-76, January 1988.

[98] K. Shikano, K. Lee, and R. Reddy, "Speaker Adaptation Through Vector Quantization", *IEEE In-*

ternational Conference on Acoustics, Speech, and Signal Processing, Paper 49.5, 1986.

[99] R. Stern, F. Liu, Y. Ohshima, T. Sullivan, and A. Acero, "Multiple Approaches to Robust Speech Recognition", *Proceedings of DARPA Speech and Natural Language Workshop*, pp. 274-279, Feb., 1992.

[100] T. Stockham, T. Cannon, and R. Ingebretsen, "Blind Deconvolution Through Digital Signal Processing", *Proceedings of the IEEE,* 63(4), pp. 678-692, April 1975

[101] T. Sullivan, and R. Stern, "Multi-Microphone Correlation-Based Processing For Robust Speech Recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 91-94, April, 1993.

[102] D. Van Compernolle, "Noise Adaptation in a Hidden Markov Model Speech Recognition System", *Computer Speech and Language*, 3, 151, 167, 1989.

[103] A. Varga, and R. Moore, "Hidden Markov Model Decomposition of Speech and Noise", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 845-848, April, 1990.

[104] V. Viswananthan and C. Henry, "Evaluation of Multisensor Speech Input for Speech Recognition in High Ambient Noise", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 85-88, April, 1986.

[105] A. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory,* vol. IT-13, pp. 260-269, 1967.

[106] W. Ward, "Modelling Non-verbal Sounds for Speech Recognition", *Proceedings of Speech and Natural Language Workshop*, pp. 47-50, Oct. 1989.

[107] B. Widrow, and J. Glover, "Adaptive Noise Cancelling: Principles and Applications", *Proceedings of IEEE*, vol. 63, pp1692-1716, 1975.

[108] B. Widrow, and S. Stearns, *Adaptive Signal Processing*, Prentice-Hall Inc., 1985.

[109] J. Wilpon, L. Rabiner, C. Lee, and E. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models", *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 38, No. 11, pp. 1870-1878, Nov. 1990.

[110] P Woodland, J. Odell, V. Valtchev, and S. Young, "Large Vocabulary Continuous Speech Recognition Using HTK", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 125-128, April, 1994.