Multi-Microphone Correlation-Based Processing for Robust Automatic Speech Recognition

by

Thomas M. Sullivan

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

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

August 1996

Abstract

Speech recognition systems suffer from degradation in recognition accuracy when faced with input from noisy and reverberant environments. While most users prefer a microphone that is placed in the middle of a conference table, on top of a computer monitor, or mounted in a wall, the recognition accuracy obtained with such microphones is generally much worse than the accuracy obtained using a close-talking headset-mounted microphone. Unfortunately, headset-mounted microphones are often uncomfortable or impractical for users. Research in recent years on environmental robustness in speech recognition has concentrated on signal processing using the output of a single microphone to correct for differences in spectral coloration between microphones used in the training and testing environments, and to account for the effects of linear filtering and additive noises present in real testing environments. This thesis explores the use of microphone arrays to provide further improvements in speech recognition accuracy.

A novel approach to multiple-microphone processing for the enhancement of speech input to an automatic speech recognition system is described and discussed. The system is loosely based on the processing of the binaural hearing system, but with extensions to an arbitrary number of input microphones. The processing includes bandpass filtering and nonlinear rectification of the signals from the microphones to model the effects of the peripheral auditory system, followed by cross-correlation within each frequency band of the outputs of the rectifiers from microphone to microphone. Estimates of the correlated energy within each frequency band are used as the basis for a feature set for an automatic speech recognition system.

Speech recognition accuracy in natural environments and using artificially-added noise were compared using the new correlation-based system, conventional delay-and-sum beamforming, and traditional adaptive filtering using the Griffiths-Jim algorithm. It was found that the more computationally-costly correlation-based system provided substantially better recognition accuracy than previous approaches in pilot experiments using artificial stimuli, and in experiments using natural speech signals that were artificially corrupted by additive noise. The correlation-based system provided a consistent, but much smaller, improvement in recognition accuracy (relative to previous approaches) for experiments conducted using speech in two natural environments. It is also demonstrated that the benefit provided by microphone array processing is complementary to the benefit provided by single-channel environmental adaptation algorithms such as codeword-dependent cepstral normalization, regardless of which adaptation procedure is employed.

Acknowledgments

It has been a long road. I've met, worked with, and been influenced by so many people along the way that it will be very hard to make sure to get them all in.

First and foremost, I have to heap my thanks on my advisor, Dr. Richard M. Stern. Rich was more understanding and supportive along my graduate school trek than I deserved. He wrote sterling recommendation letters to help me get into graduate school, get job offers, etc. He took me in here at CMU when things were rough for me at MIT, he gave me a project when the funding for my initial music work here was ended. He gave me plenty of room when I wanted to pursue my other interests, all of which resulted in this thesis taking much more time than it should have. He funded me from slush funds when my corporate funding had run out. I can never repay him for all he's done for me, but I will also never forget it and will appreciate it forever.

Many thanks go out to Motorola Corp. for funding a major portion of this research, and especially to Ira Gerson and Bill Kushner of Motorola, who were assigned to oversee my work for the duration of the support. I'd also like to thank ARPA, who funded portions of this research as well.

I have to thank the members of our group and my office mates over the years. They've made the Porter Hall area very sociable and intellectually stimulating. Alex Acero, Alexei Sacks, Vipul Parikh, Fu-Hua Liu, Pedro Moreno, Matt Siegler, Nobu Hanai, Uday Jain, Bhiksha Raj, Evandro Gouvea, Juan Huerta, Sam-Joo Doh, Adam Liss, and Sammy Tao. Most of all, I'd like to thank Aki Ohshima. Aki was my office mate, bandmate, drinking buddy, colleague, and still is one of my best and closest friends. Much of my thesis software has many hours of his toil imbedded within as well, to which I owe him much gratitude.

I owe my undying gratitude to my girlfriend Kris, who came along just before my thesis proposal and has been with me and supporting me ever since. She's had to see it all, and has always been there for me when I've needed her. I love her so much and I hope I can repay her love as well for the rest of both of our lives.

I'd also like to thank the members of the CMU Hockey Club. This group of folks has been my biggest outlet for stress relief for the past 6 years. I've enjoyed every minute of every practice, game, carnival and party with these folks. They helped make the dream of playing competitive ice hockey (that I carried around with me since I first fell in love with the sport at age 7) a reality.

I'd like to thank my other committee members, Dr. Raj Reddy, Dr. Jose M.F. Moura, Dr. Virginia Stonick, and Dr. Wayne Ward for taking the time to provide their insight, time, and constructive criticism to this work.

And I cannot possibly end this without thanking all of my family members. My brothers and sisters Sue, John, Cheri, Kevin and Am. My step- brothers and sisters John, Mary, Kathy and Matt. All of their young'uns (my nieces and nephews). And especially my stepfather Joe Schiffgens who has really come to bat for us all in times both good and bad.

This thesis is dedicated to:

My father John V. Sullivan (died May 16, 1975), one of the brightest and most caring people I've ever known. I wish I'd had the benefit of his insight and innovation into my adulthood.

My mother Geraldine J. (Gottus) Sullivan (died July 26, 1984), a special education teacher and one of the most honest and positive people I've ever known. I wish she were here to see me finish my Ph.D. She worked so hard to give me this opportunity.

My grandfather Thomas Gottus (died Nov. 8, 1983), had to leave school after 4th grade to go to work to make money for his family. An amazing self-taught person who could build or repair anything he liked. It is his common sense approach to problems which has had a huge impact on me over the years.

And to my grandmother Rosalie Gottus (still going strong!). An amazing testament to never giving up on things you like to do. I hope I have her longevity and energy when I reach her age.

I couldn't have asked for better parents and grandparents. These people believed so highly in education and providing education to their children that without their support and example I'd never have had the opportunity to get to this point today. I hope I can carry on this devotion to education throughout my own life.

Chapter 1. Introduction

The performance of an automatic speech recognition system suffers in environments corrupted by noise sources. In systems where a close-talking microphone (such as a headset) is used, the system is isolated from these effects to a large degree. But such microphones are undesirable and even impractical in many environments. Close-talking microphones (if wired) require the user to be physically connected to the system. Headsets may be uncomfortable for the user or obstruct necessary hearing or vision. Methods are required for robust automatic speech recognition systems to make the system performance when using desktop or other room-mounted microphones comparable to the performance when using a close-talking headset microphone.

Speech signals in rooms can be colored by a variety of noises. These noise sources can be uncorrelated additive noises (computer and air-conditioning fans, rustling papers, coughs) or correlated noises (such as echoes due to room reverberation). These corrupting sources may also be localized or spatially scattered. Localized sources include competing speakers, nearby computer fans, etc., whereas spatially scattered noises would include copies of a desired signal due to room reverberation, fans from climate-control systems, etc.

1.1. The Cross-Condition Problem

The problem of poor recognition results in cross-conditions has motivated research in environmental robustness for speech recognition systems. Table 1-1 provides some baseline comparisons of the recognition accuracy (word correct) obtained using CMU's SPHINX-I speech recognition system trained and tested using both a close-talking (CLSTK) microphone and a desktop (Crown PZM6FS) microphone in an office environment [Acero, 1990]. The task was an alphanumeric census task where the utterances were strings of digits, letters, and basic one-word database commands ("no", "enter", "stop", etc.).

As we see from the table, the cases involving the Crown PZM6FS microphone exhibit worse performance for a given testing condition than the close-talking cases. Note especially the poor performance for cross condition cases, *i.e.* for cases for which the system was trained on speech collected with a close-talking microphone and tested on a desktop microphone, and vice versa. A robust system would be one in which all conditions yielded similar performance to that observed when the system is trained and tested using the close-talking microphone.

	Test CLSTK	Test PZM6FS
Train CLSTK	85.3%	18.6%
Train PZM6FS	36.9%	76.5%

Table 1-1: Baseline SPHINX word correct accuracy results for various training/testing conditions.

A degree of success in correcting the discrepancy due to dissimilar training and testing conditions has been gained from environmental normalization algorithms that work on monophonic speech signals such as Acero's CDCN (Codeword Dependent Cepstral Normalization) algorithm [Acero, 1990]. Such algorithms provide a normalization between the training speech environment and the testing speech environment such that the system sees them as more equivalent. This works well, but as will be shown in Chapter 4, better results can be achieved when array processing is used in conjunction with CDCN in the recognition system's front end.

The benefit that array processing systems can provide is an increase in signal-to-noise ratio (SNR) due to multiple-microphone inputs which may be combined and processed in various ways. Figure 1-1 shows a microphone array receiving two signals. The "desired signal" arrives at the array on-axis, meaning the signal arrives at all microphones in the array at the same time. The "jamming signal" arrives off-axis. It will therefore arrive at a slightly different time to each of the sensors. It is this time delay that we wish to exploit to reject the jamming signal(s), or to emphasize the desired signals. Arrays may also be steered in the direction of desired sources and nulls placed in locations of undesired localized sounds, but adapting to diffuse sounds is much more difficult.

From the human factors standpoint, a motivation for our study of the application of array processing techniques to speech recognition systems is the desire to provide a more ergonomically acceptable environment for speech recognition.

Three major categories of array processors have been used in speech recognition: delay-and-sum arrays, traditional adaptive arrays, and arrays using correlation-based processing. These will be described in more detail in Chapter 2 along with some other recent systems.

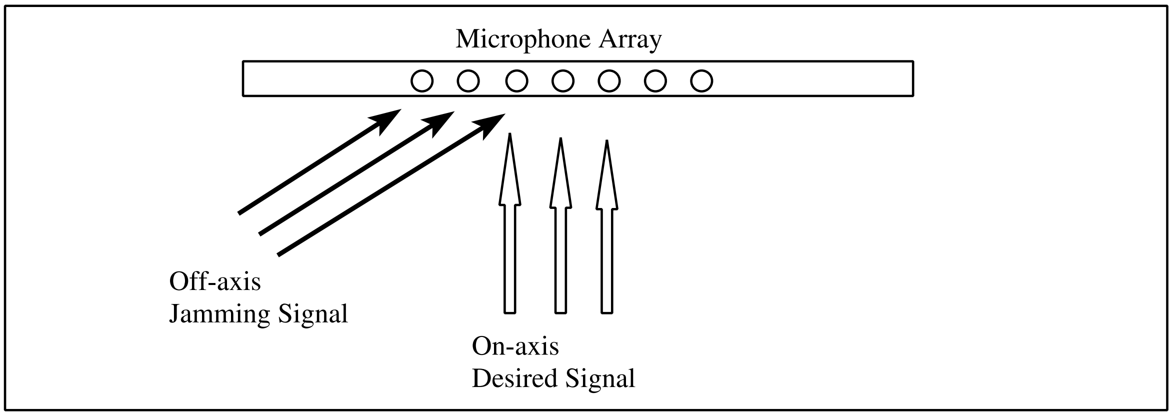

Figure 1-1: An on-axis "desired signal" and off-axis "jamming signal" arriving at a microphone array.

1.2. Thesis Statement

We propose a novel approach to the robustness problem based on the cross-correlation between multiple microphone channels. This approach is motivated by the cross-correlation processing that takes place in the human binaural system, which is known to be very robust in a wide variety of environments as well as providing us a means for localizing sounds. Up to now, cross-correlation processing has been used primarily in systems designed to determine the localization of signals, rather than for improving the quality of input for speech recognition.

The goal of this research has been to use an array of microphones to provide the means for sound localization and the enhancement of a cepstral feature set derived from speech signals for an automatic speech recognition system. We chose as our testing environments a laboratory that is heavily corrupted with many computer fans and disk drives, and a small conference room.

1.3. Thesis Overview

In Chapter 2 of this thesis we provide some background into noise reduction and enhancement of speech signals, and we discuss some of the existing types of array processing algorithms. We discuss the advantages and disadvantages of the various approaches to array processing for speech recognition, and we argue why we feel the correlation-based approach is the method best suited to further exploration. We also discuss our goals for a successful correlation-based system.

Chapter 3 presents the SPHINX-I automatic speech recognition system used for the recognition experiments in our work.

Chapter 4 presents an initial experiment that was carried out in collaboration with the CAIP Center at Rutgers University to demonstrate the potential gain in recognition word accuracy by incorporating array processing methodology with an automatic speech recognition system.

In Chapter 5 we present our multi-microphone correlation-based algorithm and we discuss its implementation.

Chapter 6 presents a series of pilot experiments we have performed to demonstrate the feasibility of the correlation-based processing algorithm which is the focus of this thesis. Experiments using artificially-generated stimuli and actual speech segments are described.

Chapter 7 describes a series of experiments performed to evaluate the performance of our algorithm in the context of realistic speech recognition tasks. We also present some comparisons to some of the traditional array processing methods discussed in Chapter 2.

Chapter 8 will present some further discussion of the array performance and describe some areas of further research that would help us to extend the usefulness of our algorithm. These areas may also provide greater understanding of the system such that future array processing algorithms for speech recognition systems will continue to improve the performance of these systems.

Chapter 2. Background

In this chapter we discuss approaches to signal enhancement which incorporate multiple microphone inputs. We will concentrate on three major methods; delay-and-sum beamforming, traditional adaptive arrays, and correlation-based array processing. The major differences in the methods are in the way the multiple input signals are combined and then processed. This determines the type of overall response to the incoming signal one obtains from the array. We will point out the advantages and disadvantages of each approach, and provide some examples of research done using each type of system.

In the discussions of methods and experiments in the remainder of the chapter, it is generally assumed that the direction of the desired signal is known, and the array will be aligned in that direction. Discussion on localization of signals will take place later in the thesis.

2.1. Delay-and-Sum Beamforming

In delay-and-sum beamforming, delays are inserted after each microphone to compensate for the arrival time differences of the speech signal to each to each microphone (Figure 2-1). The time-aligned signals at the outputs of the delays are then summed together. This has the effect of reinforcing the desired speech signal while the unwanted off-axis noise signals are combined in a more unpredictable fashion. The signal-to-noise ratio (SNR) of the total signal is greater than (or at worst, equal to) that of any individual microphone's signal. This system makes the array pattern more sensitive to sources from a particular desired direction.

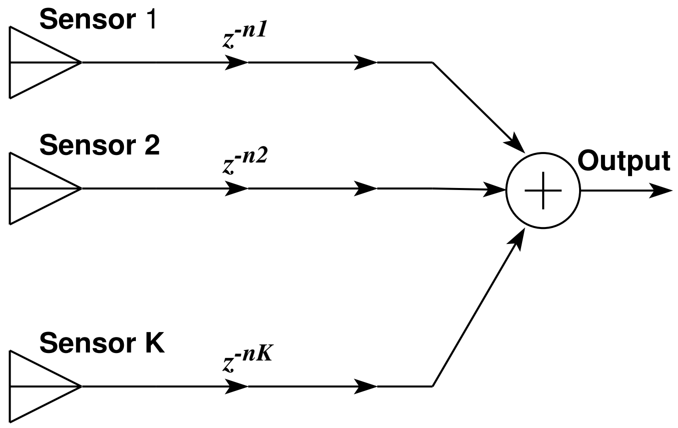

Figure 2-1: Delay and sum beamformer.

The major disadvantage of delay-and-sum beamforming systems is the large number of sensors required to improve the SNR. Each doubling of the number of sensors will provide at most an additional 3 dB increase in SNR, and this is if the incoming jamming signals are completely uncorrelated between the sensors and with the desired signal. Another disadvantage is that no nulls are placed directly in jamming signal locations. The delay-and-sum beamformer seeks only to enhance the signal in the direction to which the array is currently steered.

2.1.1. Application of Delay-and-Sum Processing to Speech Recognition

Delay-and-sum processing is attractive because of its simplicity. It is easy to implement, provides some improvement, and can be done is a very cost-effective manner. Flanagan *et al.*, [1985] developed one of the first practical systems based on delay-and-sum beamformers. The original systems were designed at AT&T and consisted of a linear array of sensors. Steering delays were applied using analog bucket-brigade technology, and the summed output signals provided the enhanced speech. They also made two-dimensional versions of the arrays to be used in lecture halls. More recently [Kellerman, 1991], completely digitally processed versions were built. Flanagan [1994] has also proposed three-dimensional architectures which could be placed along chandeliers in conference rooms, etc.

Silverman *et al.* [1992] also use delay-and-sum systems for their localization work. Hardware to correlate output signals from a pair of linear sensor arrays placed along the “x” and “y” directions in a room is used to locate the position of a desired speaker, and the outputs of the sensors are summed after the correct steering delays are implemented.

In Chapter 4 we will describe a pilot experiment we performed to confirm the benefit of delay-and-sum beamforming (and array processing in general) in speech recognition applications.

2.2. Traditional Adaptive Arrays

Traditional adaptive arrays process the incoming microphone signals each through its own tapped delay line with variable weights. The outputs of the individual delay lines are then summed into a single output. The optimal set of weighting coefficients is determined by minimizing the squared error between this output signal and the desired response. These are called Minimum Mean Squared Error (MMSE) systems.

While delay-and-sum arrays attempt to enhance the desired signal by steering the beam in the direction of the desired signal, traditional adaptive array algorithms (such as the Griffiths-Jim and Frost algorithms [Widrow and Stearns, 1985]) also attempt to place nulls in the direction of undesired or "jamming" signals by minimizing the mean squared error between the energy of the desired signal and that of the processed signal from the array (Figure 2-2). (The delay-and-sum beamformer can be thought of as a degenerate case of traditional adaptive array processing with a single unweighted delay per channel inserted to simply align the signals. Fixed delays, as in delay-and-sum beamforming, are sometimes inserted at the input of a traditional adaptive array to "steer" the beam in a desired direction.)

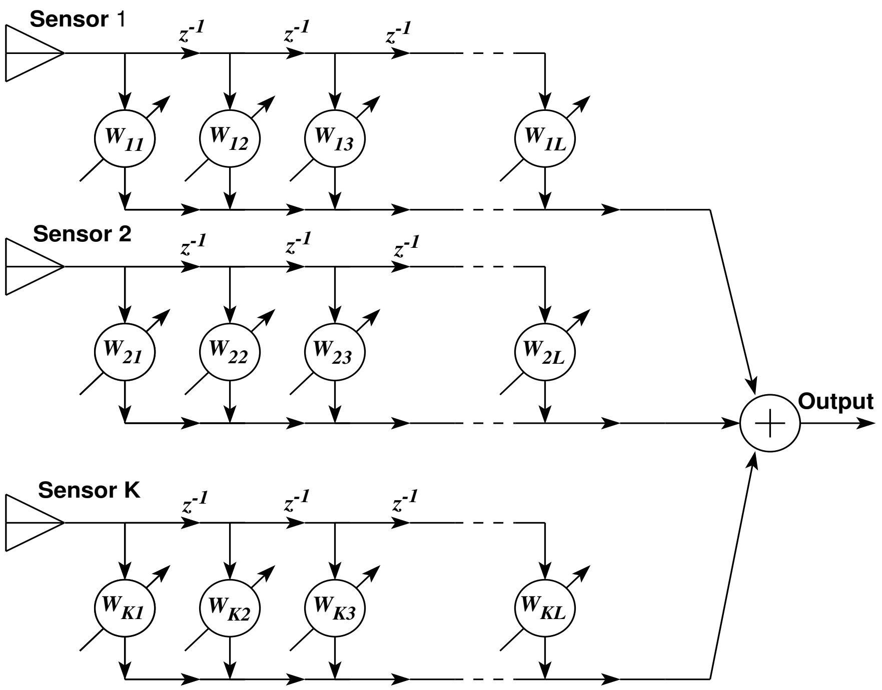

Figure 2-2: Traditional Adaptive Array.

Since MMSE methods assume that the jamming signals are statistically independent from the desired signal, they can suffer from signal cancellation due to the correlated nature of the jammer signals in reverberant environments (copies of the desired signal arrive at the array sensors "off-axis"). For example, Peterson [1989] observed very poor response using the Griffiths-Jim algorithm in a reverberant room, compared to the performance obtained using the same algorithm in an anechoic chamber. (One way to avoid this signal cancellation is to adapt the processor filter coefficients only when noise is present, *i.e.* perform the adaptation during the non-speech segments of the signal, as was demonstrated by Van Compernolle [1990] and will be discussed in more detail in a later section.)

Secondary reflected components of speech signals in reverberant environments appear very shortly after the primary component. This makes echo-cancellation techniques such as those used for adaptive equalization over telephone lines difficult to use for speech recognition because the window for adaptation is much shorter [Sondhi and Berkley, 1980].

2.2.1. Adaptive Noise Cancelling

One of the most basic uses of traditional adaptive filtering methods is in Adaptive Noise Cancelling (ANC) [Widrow et. al. 1975]. In ANC, a reference sensor is used to provide a means of collecting a sample of noise one wishes to cancel from the primary sensor, which contains both the desired signal and the noise. The reference noise is correlated to the noise in the primary channel.

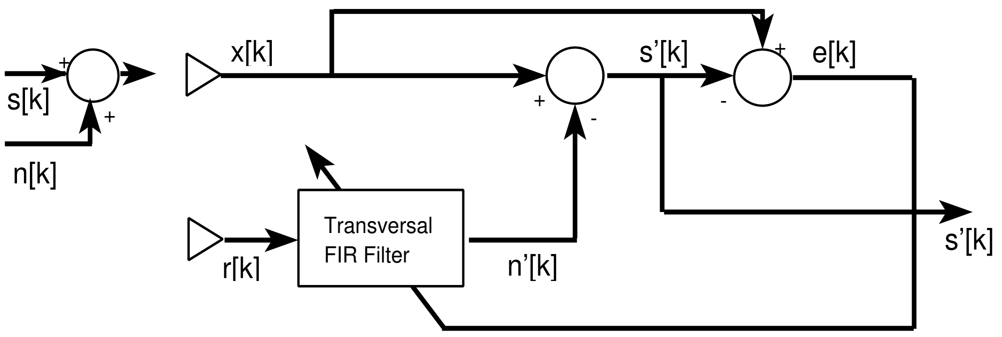

Figure 2-3: Adaptive Noise Cancellation system.

Figure 2-3 shows a basic ANC system. Here $s[k]$ is the desired incoming signal and $n[k]$ is the corrupting noise, with $x[k]$ being the sum of the two which is the signal that appears at the primary sensor. Signal $r[k]$ is the reference signal, placed away from $x[k]$ such that it collects a noise sample that is correlated to $n[k]$. The reference signal $r[k]$ is then filtered with an FIR filter to provide a noise signal as similar as possible to $n[k]$. The filter simulates the environment or room response difference between the noise at the primary sensor and the reference sensor. The taps of the filter are adapted such that the energy of the error signal $e[k]$ is minimized. The recovered signal, $s'[k]$, is ideally very close to the original $s[k]$.

The ANC works well if the reference signal does not contain much (if any) of the desired signal. If the desired signal is present in the reference signal, then the system will attempt to adapt to the desired signal as well, and will cancel it. This signal cancellation is a potential problem in any MMSE based system.

Another problem that may arise in using ANC systems for broadband signals is the length of the filter. The filter may need many taps if it is to provide a good representation of the spectral variations in the environment. Increasing the size of the filter causes the adaptation to be slower in the system. If the noise source is non-stationary, this may lead to decreased performance of the canceller due to its inability to track fluctuations in the noise signal.

The ANC-based systems have been applied to speech enhancement by many researchers. *(e.g.* Widrow [1975], Widrow and Stearns [1985], and Boll and Pulsipher [1978]). Generally they are used to cancel additive noise sources from degraded speech. For reverberant environments, the signal is present as a jamming signal due to room reflections, and the signal is also present in every sensor.

There is a huge body of work using traditional MMSE-based beamforming algorithms for speech enhancement. Beamformers work by processing each of the sensor inputs such that a desired spatial response pattern is attained. Narrowband beamformers are the simplest in that they simply use a weighted sum of the input sensors. This allows one to optimize the array response for a particular frequency by choosing the set of weights that gives the desired response for the operating frequency (i.e., the jammer signal is a tone at a particular frequency). For broadband signals, a tapped delay line (FIR filter) is required to process each sensor prior to combing them. The weights of the filter taps can be adaptively changed such that the error between the desired signal and attained signal is constantly minimized.

2.2.2. Application of Traditional Adaptive Methods to Speech Recognition

In recent years there has been increased interest in the application of adaptive noise suppression methods and adaptive array processing methods toward improving the accuracy of automatic speech recognition systems.

Vea [1988] used a version of the Frost algorithm [Frost 1972] to conduct speech enhancement experiments (Figure 2-4). The Frost algorithm places a hard constraint on the beamformer's response by constraining the look-direction response of the array to a particular set of weights. This is accomplished by forcing the sum of the corresponding tap weights from each sensor to equal a pre-determined value. The output power is now minimized subject to this constraint. If no constraint were imposed, all of the filter weights would tend toward zero, which always produces minimal output power.

Vea found signal cancellation to be a problem for all of his processed signals. He found the Frost algorithm to be ineffective in reverberant environments.

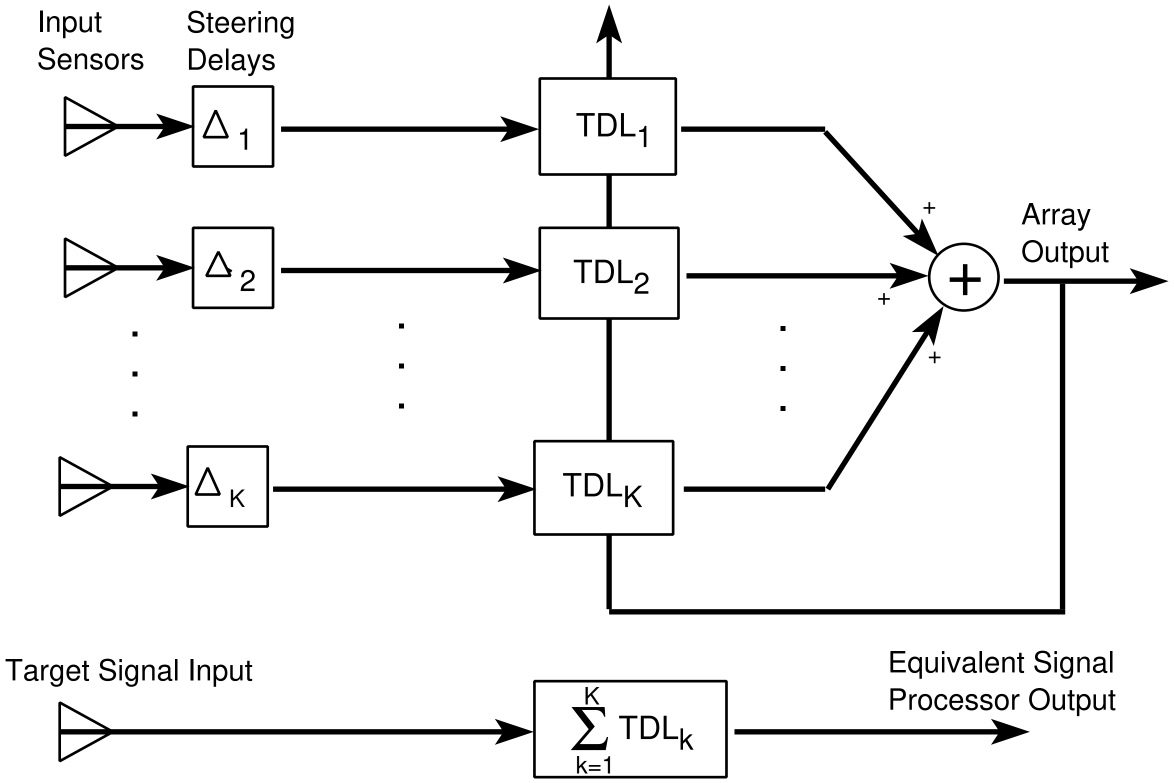

Figure 2-4: Frost algorithm

More recently, signal enhancement work using adaptive arrays for speech signals was evaluated by Peterson [1989]. Peterson used a two-sensor version of the Griffiths-Jim algorithm [Griffiths 1982]. The Griffiths-Jim algorithm (Figure 2-5) uses a delay-and-sum beamformer to form the desired signal (still corrupted by noise), and subtracts successive pairs of sensors to form the noise signals. The noise signals are then filtered and subtracted from the desired signal. The weights of the filter taps are adapted to minimize the output error, which is then closely related to the clean desired signal. Peterson had mixed success in nulling out jamming signals in three conditions; an anechoic chamber (good response), a living room (poorer response), and a reverberant room (poor response). The more reverberation present in the environment, the more difficulty his algorithm had in placing nulls at the locations of the undesired signals.

As stated previously, signal cancellation is a problem for LMS algorithms in reverberant environments. Van Compernolle [1990] avoided the problem of signal cancellation by only adapting the weights of the noise signal filters during purely noisy portions of an incoming speech signal. He detected speech and non-speech portions of the input signal and froze the weights for the filters during the speech portions of the inputs so that signal components would not be present during adaptation. He obtained some improvement in recognition accuracy by using the Griffiths-Jim processing, and by adapting only during noisy portions, which reduced the signal cancellation. His system used four input sensors.

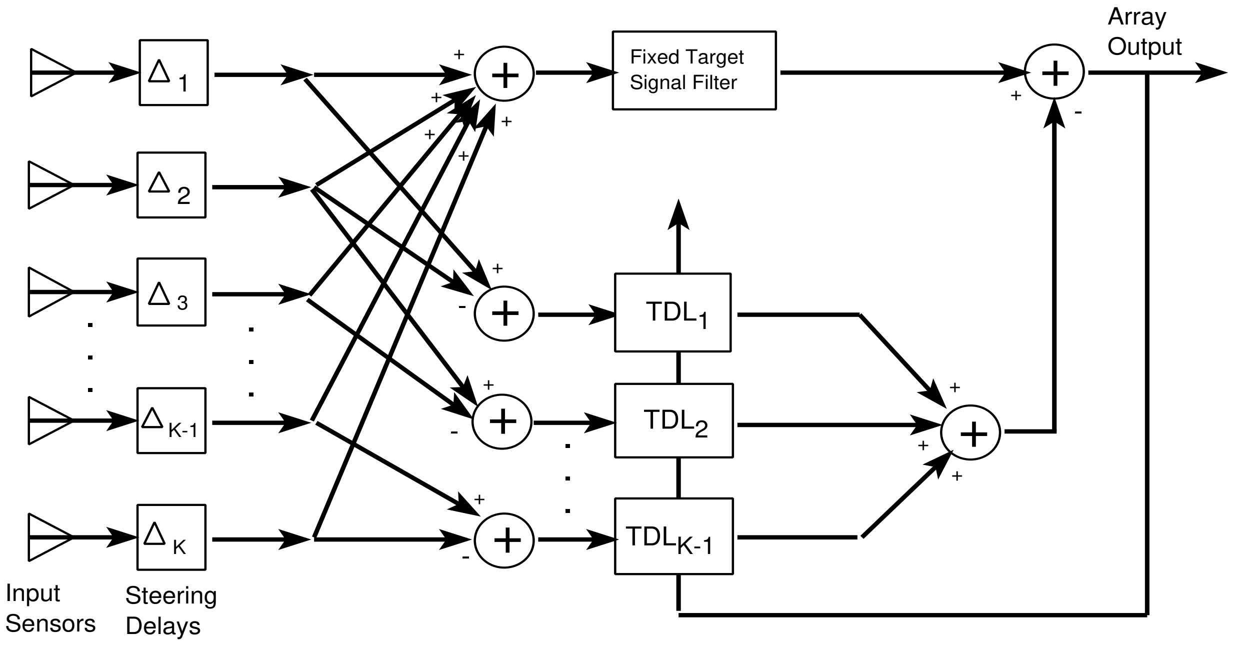

Figure 2-5: Griffiths-Jim algorithm

2.3. Cross-Correlation Based Arrays

Cross-correlation-based arrays differ from traditional adaptive arrays in that the outputs of taps along a delay line are multiplied together, rather than added, as in the block diagram in Figure 2-6. Further processing is generally performed only on the signal appearing at a particular output of the correlation array representing the desired signal. For instance, if the desired signal is delayed in time between each input sensor, the output signal containing the most energy (highest peak in the correlation display) would be the one to be processed further.

The human binaural system is an example of a two-input cross-correlation-based processing system that is known to work quite well in the presence of noise and reverberation. Our auditory system also performs well in the presence of competing talkers (the "cocktail party effect"). The human auditory system can be modeled (in part) by a pair of input sensors (our ears), a filterbank (representing the frequency selectivity of the basilar membrane in the cochlea), non-linear rectification (representing the processing of the hair cells that generate electrical impulses from the mechanical motion of the basilar membrane), and interaural cross-correlation of the outputs of the rectifiers. The cross-correlation enables incoming signals to be localized by finding the peak in the cross-correlation function between the input signals. In addition, the human auditory system exhibits a phenomenon known as the precedence effect in which the initial onset component of a signal is "recognized" by the lateralization mechanism and latter components (at least in the near term) are inhibited. This inhibition mechanism helps us to recognize sounds in reverberant environments by suppressing short-term echoes from incoming signals.

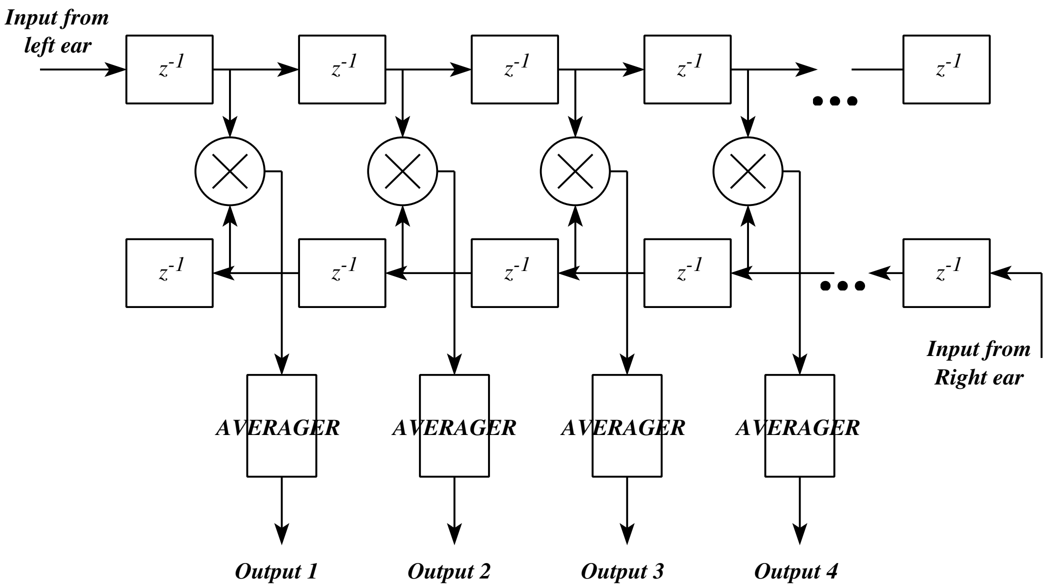

Figure 2-6: Cross-correlation processing.

2.3.1. Phenomena

2.3.1.1 Localization

The human binaural system is excellent at estimating the originating location of an incoming signal within our environment. By utilizing the delay in arrival time of a signal to each of our ears, we can estimate (to a good degree) the direction of the signal's source. Of course we use visual cues to aid us as well, but even with our eyes closed, we can localize to a high degree of accuracy. Much research has been done on how humans localize sounds, and using multiple sensors to construct localization systems.

2.3.1.2 Competing Sources (the Cocktail Party Effect)

Humans are also very good at focusing on a particular speaker or sound source in a room where many competing sources are present. Even when the competing sources are at a higher level than the desired source, humans can remain focused on the desired source. While it is still a very difficult problem to build systems that separate competing signals, it is apparent that having the spatial input cues as a result of our two ears aids us in this process. Signal separation is more difficult for humans if only one ear is used for input, because the localization cues are lost.

2.3.1.3 The Precedence Effect

The human binaural system is able to localize sounds even in environments where there is a high degree of reverberation. Reverberation is the result of delayed versions of the original signal arriving to our ears from different directions due to reflecting surfaces in our environment. This suggests that humans have a mechanism in place to focus on the first or most direct occurrence of an input signal and that we are able to suppress the delayed and reflected portions. This is known as the "precedence effect". Blauert [1983] studied the mechanism involved in the precedence effect, noting that it operates on reflections up to about 10 ms and is present in each ear individually (works in monaural situations) as well as co-laterally (binaural).

2.3.2. Binaural Models

Jeffress [1948] proposed the earliest cross-correlation model of the human binaural system to explain localization of sound sources. His model included a hypothetical neural framework in which arrival time differences of stimuli to our two ears, or interaural time differences (ITDs), were converted into "place information" for localizing the stimuli. Stevens and Newman [1936] had found that for frequencies less than 1500 Hz, tones could be localized based on the time (or phase) difference of the stimulus from the two ears. Jeffress proposed a model in which the time it took a nerve impulse to travel through a nerve fiber in the ear was related to the length of the fiber. He proposed that these fibers were "tapped" by smaller fibers, such that the location of the signal along these smaller fibers was related to the location of the signal in space.

By providing an interconnection between the fibers in the left and right ears, Jeffress' model was the first to propose a cross-correlation type mechanism. Observing which fibers from each ear fired coincidentally, the model provided a means of estimating the location of a signal in space, as it would arrive to one ear at a time delay with respect to the other. He also hypothesized that the same mechanism could be used for localizing frequencies above 1500 Hz which are detected by level differences between the two ears (interaural level differences, ILDs) by looking at the level differences at the outputs of the smaller fibers.

Stern and Colburn [1978] elaborated on the Jeffress model. Their model describes a mechanism for generating an estimate of the internal cross-correlation function which the nerve fibers calculate. The model describes the nerve firing rates by statistically independent Poisson processes. They also extended their model by proposing a mechanism that generates a position variable by combining the outputs of the binaural cross-correlator with an intensity function that depends on the interaural level differences of the input stimulus.

Lyon [1983] described a system for localizing and separating incoming signals. His model used a bandpass filterbank, halfwave rectification and some amplitude compression of the signals prior to the cross-correlation of the signals. He then looked at the peaks in the correlation functions and attempted to group peaks with similar signal phase, onset time, envelope, etc., to localize and separate the signals. Drawbacks to his methods were that the "correct" peak of the correlation function was sometimes not the largest peak (which would be the case if a competing speaker at a different location was actually speaking louder than the desired speaker), and his methods of determining which peaks to combine were somewhat *ad hoc*. However, the amount of information present in Lyon's cochleograms demonstrated that much information can be obtained about the original signals by placing it in this domain via correlation processing.

Lindemann [1986] proposed a binaural model (Figure 2-7) for the localization of sound which incorporated an inhibition mechanism. Localization of a signal was enhanced due to a sharpening of the desired signal, and a suppression of other "images" of the signal. The system is basically a pair of delay lines (one for each ear) with multiplicative operations at the outputs of each tap to simulate the cross-correlation operation. The inhibition mechanism causes the outputs of taps near a major peak in the cross-correlation function to be suppressed, thus sharpening the response (in the short term) at that particular channel.

Cross-correlation based processing, while attractive for speech recognition because of its similarity to processing by the human binaural system, is computationally expensive. The filterbanks are especially costly to implement using conventional techniques, so a formidable improvement in recognition accuracy would be necessary to make this type of processing attractive for real-time speech recognition systems. We will discuss this further later in this thesis.

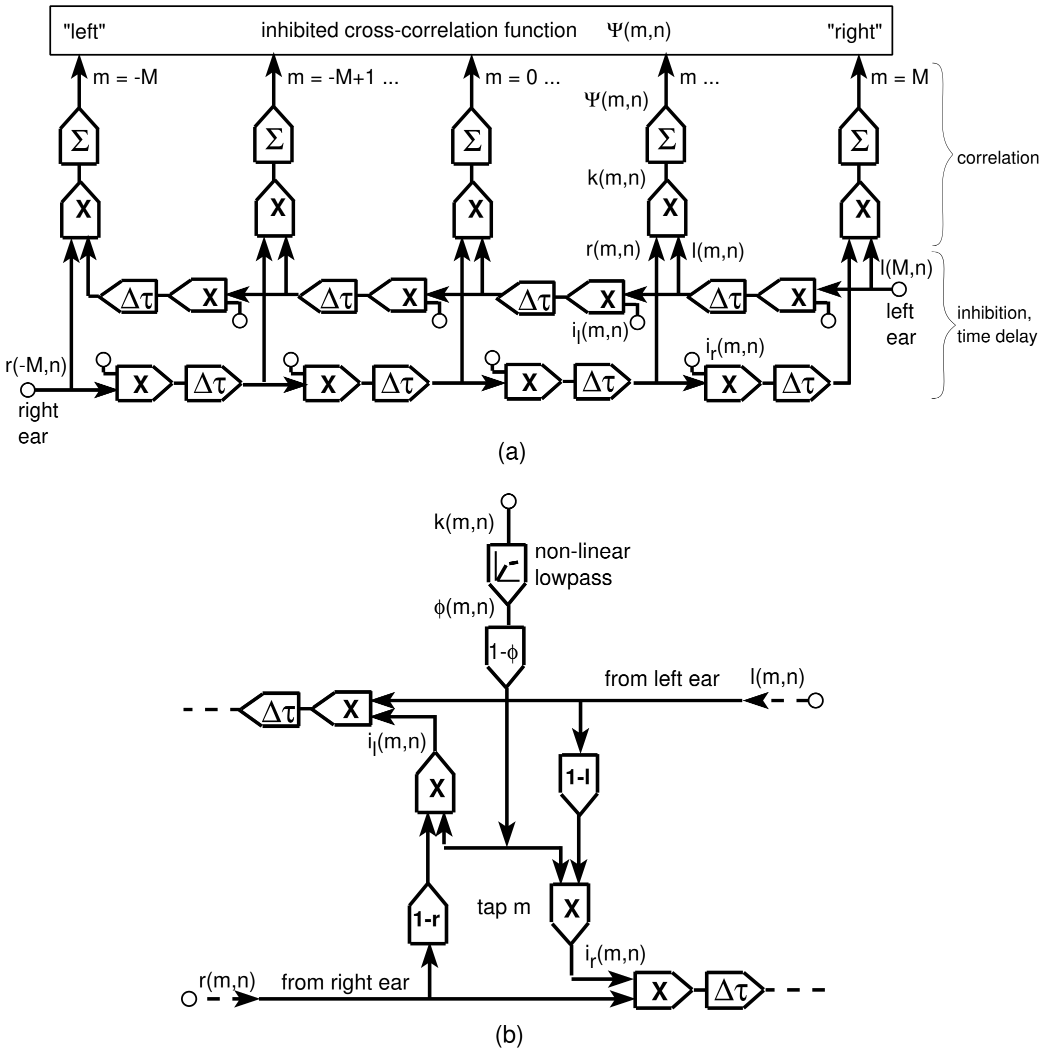

Figure 2-7: Lindemann inhibition model (a) central cross-correlation module (b) inhibitor units.

2.4. Other Multi-Channel Processing Systems

2.4.1. Binaural Dereverberation

An early system based on binaural processing to remove some of the perceived distortion from signals corrupted by reverberation was proposed by Allen *et al.* [1979]. They used a pair of microphones and passed each microphone signal through a filterbank to divide the signals into frequency bands. They then time-aligned the signals within each band by phase aligning the two signals and summing them. The gain within each band was normalized based on how correlated the two signals were within each band. The set of summed and weighted output signals from frequency bands were then summed to yield the processed output signal.They claimed that the system was effective on early room echoes and reverberant tails. Later, Bloom [1980] found that this dereverberation process had no real effect on recognition scores even though the measure of reverberation (average reverberation time and perceived reverberation time) after the process was decreased.

2.4.2. Binaural Processing Systems

Palm [1989] attempted to incorporate the Lindemann model into a cross-correlation based processing system for speech recognition. Palm rectified and cross-correlated the entire speech signal which led to distortion of the spectrum of the original speech. This spectral distortion was the result of frequency doubling, frequency smearing, etc. from the sum and difference components which result when two cosine functions at different frequencies are multiplied. Because of this signal distortion, recognition accuracy using the whole signal processed by the Lindemann model was quite poor. Nevertheless, Palm's work demonstrated that the Lindemann model was successful in localizing actual speech signals in reverberant environments, and that the inhibition did in fact sharpen the pattern obtained from the output of the cross-correlation function along the internal delay axis. He also identified the major sources of distortion in his processed signals (the nonlinearity introduced by the multiplication operation in the cross-correlation and the rectification).

2.4.3. Sub-Band Multi-Channel Processing.

Itakura proposed a sub-band processing system [Yamada, Wang and Itakura, 1991] for determining filters for dereverberation of room responses. In Itakura's system, the room response between a signal source (loudspeaker) and a sensor (microphone) is determined by passing both the signal source and the received signal at the sensor through a bank of bandpass filters and using the LMS algorithm to minimize the error between the source and sensor within each frequency sub-band. The set of filters determined by the LMS algorithm can then be applied to any received signal at the sensor to remove the room response from that signal. The individual sub-band filters are first checked for minimum phase, and only those found to be invertible are used in the processing. They found most of the filters satisfied the minimum phase criterion. Those bands not satisfying the criterion are not filtered. They found that this system helped remove room reverberation in their tests.

A further application of this system [Hikichi and Itakura, 1994] used multiple receiving-room sensors and only a single source signal. The processing is the same as above only now each sub-band has a number of receiving sensor inputs for each band instead of the single receiving sensor.

2.4.4. Recent Binaural Methods

Bodden and Blauert [1992] proposed a system based on human binaural hearing to enhance speech when concurrent speakers are present (the cocktail party effect). Their system uses the Lindemann model as a binaural processor to provide azimuth information about the direction of a desired sound to a bank of Wiener filters. A Wiener filter is designed for each of 24 frequency bands representing the critical bands in the human ear. The Wiener filters provide an optimal filter for signals in the desired look direction as determined by the binaural processor.

Bodden and Anderson [1995] extended the above model to a speech recognition task for phoneme recognition by injecting artificially additive noise to clean speech. They reported a 20-dB SNR advantage over a monaural signal for a phoneme recognition task.

2.5. Monophonic Enhancement Techniques

It is useful to mention enhancement techniques that operate on a signal collected with a single microphone. These algorithms generally require much less processing than multi-channel algorithms, but they usually only work well when additive noise or spectral shaping (from different microphone frequency responses) are present. Since one cannot obtain spatial information from a monophonic signal, these techniques do not have the directional sensitivity that multi-channel algorithms can provide.

A common model for degraded input speech signals is shown in Figure 2-8. The input speech is subjected to linear filtering effects (room response, microphone shaping, etc.) and additive noise (room noises, etc.).

2.5.1. Spectral Subtraction

Spectral subtraction algorithms attempt to remove added broadband noise from a system by subtracting the spectral shape of noisy portions of a signal from portions where speech and noise are both present. Sections of the entire signal are transformed into the frequency domain. Purely noisy portions of the signal are identified and the magnitude of the noise spectrum is estimated from these frames. This representation of the noise spectrum is then subtracted from all the frames in the original signal. If the noise is stationary throughout the duration of the signal, the noise portion of the signal can be removed, leaving only the desired signal.

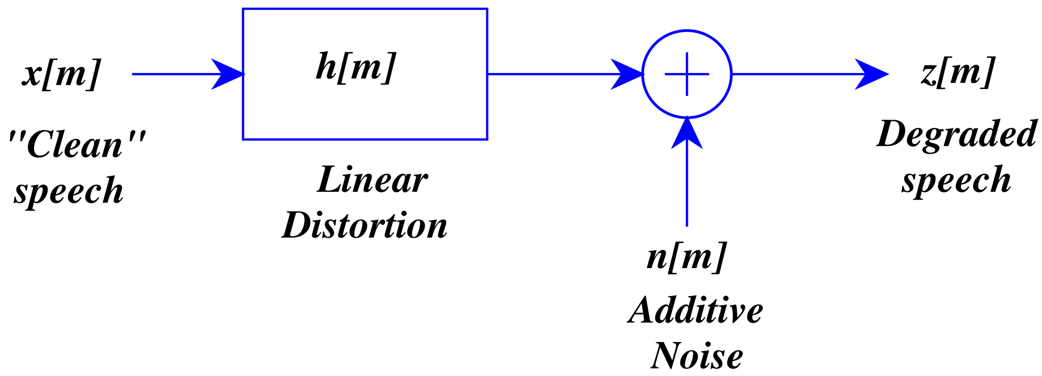

Figure 2-8: Linear filtering and additive noise model of degraded speech.

Boll [1979] used this approach for speech enhancement by estimating the noise during non-speech portions of recorded speech utterances and then subtracted this noise from the entire signal. His results yielded an improvement in that the noise in the non-speech portions decreased, but the overall intelligibility remained approximately the same. Berouti[1979] added two improvements to Boll's work by subtracting a slight overestimation of the noise and also by setting a minimum spectral level. The minimum spectral level prevented negative spectral components from being generated, something he felt caused some "ringing" in Boll's work. Once again, the intelligibility of the processed signal was largely unaffected by the processing, though listeners preferred the overall quality of the enhanced speech.

Morii [1988] implemented Boll's and Berouti's work on signals for the SPHINX system at CMU, but also attempted to correct for differences in the spectral shape of training and testing microphones. Morii was interested in speech recognition accuracy rather than speech intelligibility. He reported an overall improvement in the error rate of approximately 40%.

2.5.2. Environmental Normalization

Acero [1990] developed several methods for normalizing the cepstral vectors between the training and testing datasets in speech recognition systems. The methods allowed the system to act as if training and testing data were from the same environment, yielding superior word recognition accuracy when systems are trained and tested on data from different environmental conditions. Codeword Dependent Cepstral Normalization (CDCN) was one of the most useful products of his work.

Liu [1994] extended the cepstral normalization ideas into other similar algorithms, for the most part with good success. One algorithm Liu introduced was Fixed Codeword-Dependent Cepstral Normalization (FCDCN). FCDCN is like CDCN, only a set of correction vectors are calculated for each of a set of SNRs. Upon testing, the SNR of the frame measured, and the appropriate correction vectors for that SNR are applied to that frame.

Another algorithm that Liu introduced was Multiple Fixed Codeword Dependent Cepstral Normalization (MFCDCN), in which the FCDCN is performed independently for a set of different microphones used to collect the training data. The testing environment is then compared to the set of training environments, and the set of compensation vectors for the training environment most closely matching the testing environment is used for the compensation.

Moreno [1996] provided a set of extensions which include Multivariate-Gaussian-Based Cepstral Normalization (RATZ), Statistical Reestimation (STAR), and the Vector Taylor Series (VTS) approach. In RATZ, correction factors are applied to the incoming cepstral vectors of noisy speech. In STAR, modifications are made to some of the parameters of the acoustical distributions in the Hidden Markov Model (HMM) structure. For VTS, the adaptation is reduced to a single sentence (the sentence to be recognized) by taking advantage of extra knowledge provided by a Taylor series model of the data.

2.5.3. Homomorphic Processing

Room reverberation may be modelled by a set of acoustic impulse responses from each source location to each sensor location around a room. A reverberant signal received by a sensor would be the convolution of the original input signal with the acoustic impulse response of the room. Room reverberation can be removed in some cases (Schafer [1969] and Stockham *et al.* [1975]) by applying homomorphic filtering to deconvolve the room impulse response from the desired signal. It is necessary to model the impulse response of the actual environment and to have a good idea of the number of significant echoes present.

In more complex rooms, it is not possible to perform this type of processing to remove reverberation via homomorphic processing. For example, if the room impulse response is not minimum phase (all poles and zeros inside the unit circle), the inverse response may be unstable.

A simple method of homomorphic processing is cepstral mean normalization (CMN). In CMN, one finds the mean value of each of the cepstral coefficients over the duration of an utterance and subtracts that mean value from the cepstral coefficient vector for each frame of the utterance. This helps correct for spectral tilt and gives a large improvement in recognition accuracy at little computational cost. CMN is used in all of the speech recognition experiments described in this thesis.

2.6. Summary

In summary, delay-and-sum systems provide some signal enhancement, are very simple to implement, but the number of microphones needed for a very robust system is large (since one gains a maximum of 3 dB in SNR for every doubling of the number of microphones). Traditional adaptive arrays have the advantages of being able to place nulls in the directions of jamming signals and greater potential processing gains, but they suffer from signal cancellation problems in reverberant environments. Correlation-based systems seem to provide much information about the arrival time of incoming signals, the grouping of signals from similar sources, and they provide a possible mechanism to simulate the inhibition mechanisms needed to operate in reverberant environments.

The many monophonic processing techniques provide varying degrees of signal enhancement. Some attempt to increase the SNR by attenuating the noise portion of the signal, and others attempt to correct for reverberation and linear filtering effects.

2.7. Goals for this Thesis

The goal of this thesis work is to develop a cross-correlation-based processing algorithm that provides a greater improvement in word recognition rate from the SPHINX speech recognition system at CMU than both delay-and-sum systems and traditional MMSE-based algorithms such as the Griffiths-Jim algorithm. We would like our system to achieve better recognition accuracy than delay-and-sum systems for an equal number of input sensors, and in a way that is not adversely affected by signal effects in reverberant environments as is the case with traditional adaptive array algorithms. In order to overcome the signal-distortion problem identified by Palm [1989], we will implement a system based more on the human auditory system that separates the signal into narrowband frequency components (as had previously been done by Jeffress, Colburn, Lyon, and most other researchers in binaural perception). The signals in each frequency band are then rectified and cross-correlated with similarly filtered and rectified signals from other inputs. We also explore the combination of monophonic environmental robustness techniques such as CDCN with multi-microphone array processing techniques to determine if one out-performs the other, or if a complimentary relationship exists between the two, with additional gain to be obtained by used both forms of processing.

Chapter 3. The SPHINX-I Speech Recognition System

In this chapter we provide a description of the SPHINX-I automatic speech recognition system which is used to carry out the speech recognition experiments in Chapter 7.

3.1. An Overview of the SPHINX-I System

The SPHINX-I speech recognition system was developed at Carnegie Mellon University by Lee [1989]. It was one of the first successful speaker-independent large-vocabulary continuous-speech recognition systems. It has gone through several revisions over the years (SPHINX-III is about to be released) but the basic SPHINX-I system is adequate for our task of determining the effects of array processing on automatic speech recognition.

We will describe the front-end blocks that compose the system, paying particular attention to the blocks that our correlation-based front-end algorithm will be replacing and interfacing into. A block diagram of the system is shown in Figure 3-1.

3.2. Signal Processing

Speech recognition systems use a parametric representation of speech instead of using the speech waveform itself. The parameters form a feature set which carries the information contained in the waveform itself. Generally, this feature set is a representation of the spectral envelope of the waveform and allows the useful information in waveform to be extracted with a large reduction in redundancy and decreased input data size.

The SPHINX-I system uses a frequency-warped LPC (Linear Predictive Coding) cepstrum as its feature set. The LPC analysis is done on segments of the input speech waveform, and a set of LPC cepstral coefficients are produced to represent each segment. The original LPC processing for the SPHINX-I system is as follows:

- Input speech sampled at 16 kHz.
- A Hamming window of 320 samples (20 ms) duration is applied to a 20 ms segment (frame) of speech every 10 ms. The 20 ms frames are therefore overlapped by 10 ms.
- A highpass preemphasis filter is applied to each windowed frame. The filter has a transfer function of $H(z) = 1 - 0.97z^{-1}$.
- 14 Autocorrelation coefficients are computed for each frame.
- 14 LPC coefficients are calculated using the Levinson-Durbin recursion [Rabiner and Schafer, 1978].
- 32 LPC cepstral coefficients (LPCC) are computed in linear frequency using the standard recursion.
- The bilinear transform is applied to produce 12 frequency warped LPCC for each frame.
- An additional coefficient representing the power in each frame is also computed to make the feature set size equal to 13 for each frame.

The set of 13 LPC cepstral coefficients now represent each frame of 320 samples of the input speech waveform data, which are extracted every 10 ms. An assumption made in this frame-based analysis is that the parameters in each frame are statistically independent of the parameters in all other frames, although the frames are partially correlated in reality.

The frame-based parameters themselves are called “static features”. The SPHINX system also uses “dynamic features” which are the difference in the cepstral parameters in time. This is characterized by the first-order difference of the cepstral vectors. The energy and difference energy are also used as features in each frame.

3.3. Vector Quantization

Vector Quantization (VQ) is used to further reduce the amount of data after the LPCC extraction. VQ is a data reduction technique that maps a real vector onto a discrete symbol. It was originally designed for speech coding, but is now used heavily in speech recognition and in image coding.

A vector quantizer is defined by a codebook and a distortion measure. The “codebook” is a discrete alphabet chosen to represent all speech. First, a set of mean locations within the feature space (one for each codebook member) is computed by using an iterative clustering algorithm to divide the total set of training frame vectors into a number of groups, or clusters, equal to the number of members in the codebook. Then, the vectors in each cluster are used to calculate the mean vector for that cluster. This set of mean vectors (one for each cluster or discrete alphabet member) is what is referred to as the "codebook". In testing, each frame is then determined to be a member of the discrete alphabet codebook location which is closest in Euclidean distance to the frame's vector. The difference between the actual vector and the codebook location is called the vector "distortion". So, the frame is assigned to the cluster which has a centroid yielding the lowest vector distortion for that frame's vector.

SPHINX uses three different codebooks, one for the cepstrum itself, one for the first order difference cepstra, and one which combines the frame power and difference power into a third codebook. Each of these codebooks has 256 members, allowing each codebook location to be represented by a single byte of data. Therefore, each frame of data is now represented by 3 bytes of data, one from each codebook.

3.4. Hidden Markov Models

Hidden Markov Models (HMM) are currently the most common technology used for automatic speech recognition. Rabiner and Juang [1986] present a good review of HMMs, and Picone [1990] presents a summary of HMM applications to speech recognition.

HMMs are collections of states, and the transitions from one state to another are represented by state transitions probabilities. One may also loop and stay within the same state. In SPHINX, the minimal acoustic unit of speech recognition is a form of a phoneme. These phonemes are formed in training and are represented via HMMs as a collection of state transitions from one discrete alphabet element to another. Each particular phoneme block is characterized by two sets of probabilities, the probabilities of state transitions that are conditional on the identity of the current state, and the output probabilities that specify the conditional probabilities of output symbols given a state transition.

The incoming testing speech that we wish to recognize comes in as a string of discrete alphabet symbols after the VQ. The recognition of the speech is then a task of finding the most likely HMM given the observed discrete alphabet symbol sequence. The probabilities for each of the HMMs are assigned in the training stage.

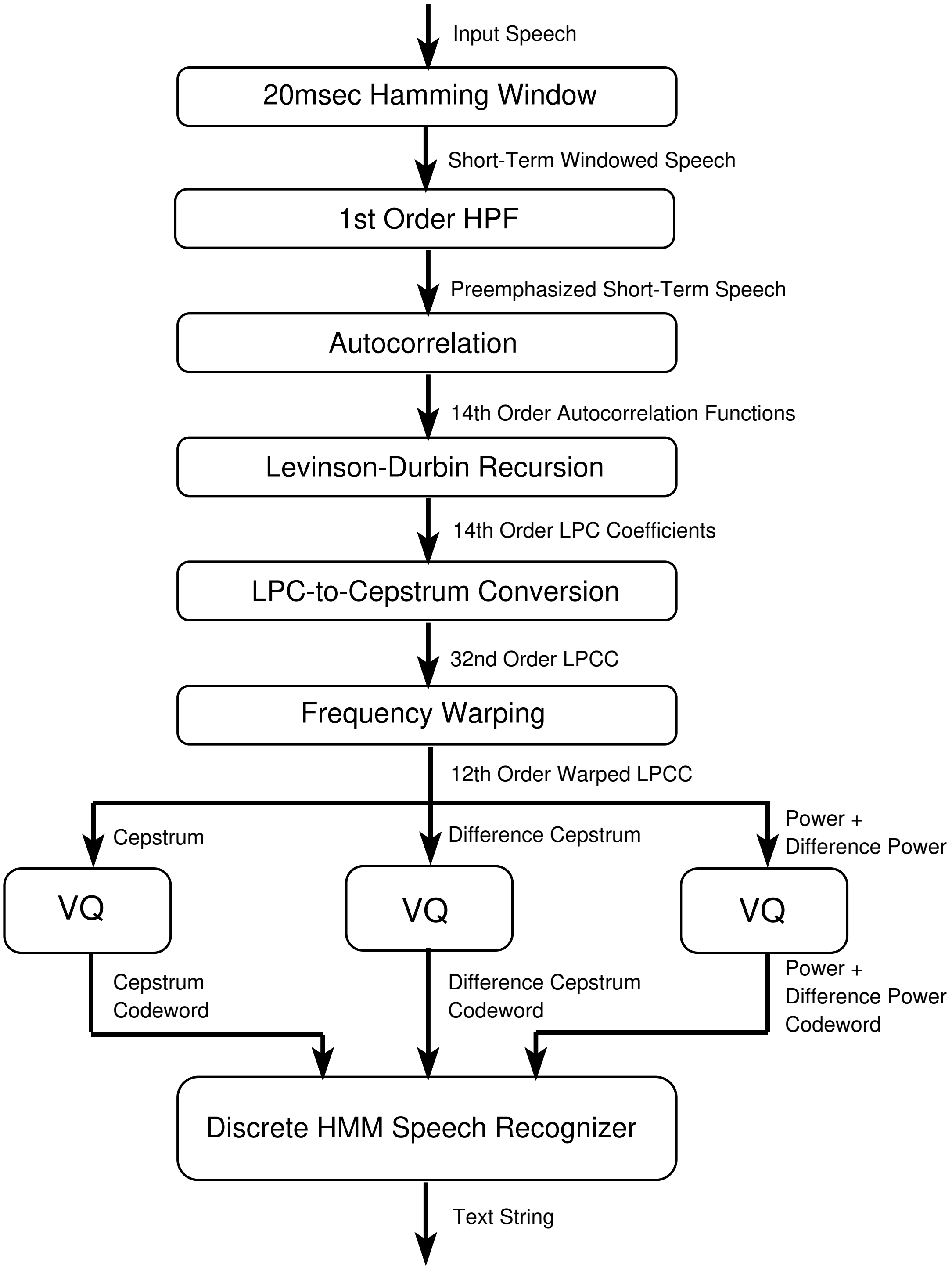

Figure 3-1: Block diagram of signal processing used in the SPHINX-I speech recognizer.

3.5. System Training

All of our speech recognition experiments were trained on the same data set, but we alternated between two different microphones in our training. The training data were collected simultaneously to the two microphones, so the transcriptions of the utterances are valid for both subsets of the data. There are 1018 utterances spoken by 74 speakers (53 male and 21 female) in the training set. The utterances were taken from an alphanumeric database consisting of strings of letters and numbers and simple data entry commands, as well as census data collected by asking the speakers to answer simple questions. Each of the speakers was asked to utter five alphanumeric utterances and was asked nine census questions. For the alphanumeric utterances, the subjects uttered the number and letter strings (ex. "XI 61") in any manner they chose or said simple data entry commands (*e.g.* "stop", "no", "enter"). The census questions (*e.g.* "Spell your name", "Say your phone number", etc.) yielded the same type of data, but were utterances that are familiar to the speaker, so the manner of speaking was expected to be more fluent.

The two microphones used in collecting the training data were the omnidirectional Crown PZM6FS microphone and the Sennheiser HMD224 closetalking headset microphone. These data will be referred to throughout the remainder of this thesis as CRPZM (for the Crown PZM) and CLSTK (for the Sennheiser close-talking microphone).

The data were collected in a small office environment (carpeted baffles were used to section off a space in a larger office room, floors were tiled, there was one computer with a fan and disk drive, etc.). The data were sampled at 16 kHz and stored as 16-bit signed integers. Data were collected in stereo with the CRPZM in one channel and the CLSTK in the other, so the set of utterances are identical for each microphone, allowing for comparisons between training microphones to be made in experiments, if desired. For this stereo database, the data collected with the CLSTK microphone had an SNR of 38.4 dB and the CRPZM data had a SNR of 19.7 dB (see [Acero, 1990] for a discussion of the exact calculation of the SNR for the alphanumeric database).

In the actual training for each experiment, we used the data from one or the other microphone to provide monophonic training data. In order to train a multi-input sensor system, the data were presented to all inputs of the system simultaneously (much like presenting a monophonic signal to both of our ears simultaneously). This allowed us to construct a feature set for the training data that was of the same form as the extracted feature set from the processing done for each of the experiments.

The data were processed in 16-ms frames (256 samples). The frames are overlapped by 8 ms (128 samples) between successive frames. (Note: baseline experiments were performed to confirm that the recognition accuracy results obtained using 16-ms frames and 8 ms of overlap were almost identical to those using 20-ms frames and 10 ms of overlap as described in Section 3.2 for the SPHINX-I LPC front-end. The value of 16 ms, or 256 samples, was chosen to allow simple FFT calculations on a full frame without zero-padding in future experiments.) Each bandpass filter output was rectified (various rectifiers have been used and will be described later). These rectified signals are multiplied by the corresponding bandpass filtered and rectified signals from the other inputs and summed over a frame length.

The recognition system for the experiments in this thesis is the three-codebook version of the CMU SPHINX speech recognition system described earlier in this chapter, using a power codebook, cepstrum codebook, and difference cepstrum codebook. Frame feature vectors are vector quantized and clustered into 256 vectors (of length 41 or 13, depending on the experiment) for the codebooks.

Upon testing, the recognition word accuracy percentage is output as the percentage of words correctly recognized by the speech recognizer. Three different types of errors are possible in the SPHINX-1 system: insertions, deletions, and substitutions. Insertions take place when the recognizer assumes a word is in an utterance that is not actually in that utterance. A deletion takes place when the recognizer assumes a word does not exist in an utterance when it actually does exist. A substitution is when the recognizer mis-recognizes a word by assuming the word is a different, incorrect word. Penalties for insertions and deletions errors can be adjusted for scoring purposes in the alignment routine that matches the recognized output of speech recognizer to the known transcriptions of the tested data. In general, it is preferable to have an almost equal balance between insertion and deletion penalties in the recognizer output scoring.

Chapter 4. Pilot Experiment Using an Existing Delay-and-Sum Array

At the time we began our research, there were no known results for systems using microphone arrays for automatic speech recognition. Microphone arrays were used primarily to locate signals, or to enhance the intelligibility or SNR of incoming signals. The purpose of the experiment described in this Chapter was to determine what, if any, performance enhancement could be obtained by using multi-microphone arrays as an input to the CMU SPHINX-I automatic speech recognizer. We also wished to evaluate whether further improvement in recognition was possible over the recognition improvement already obtained using a pre-processing algorithm such as CDCN [Acero, 1990]. We wished to determine if environmental normalization algorithms like CDCN and array processing systems complement each other or if the best possible improvement is obtained using one or the other technique.

4.1. Pilot Experiment with a Working Array

We performed an experiment at the Center for Computer Aides for Industrial Productivity (CAIP Center) at Rutgers University in June, 1991 to evaluate the performance of the delay-and-sum microphone array developed at AT&T Bell Labs [Flanagan *et al.*, 1985] and in operation at the CAIP Center. This microphone array system was developed to use multiple microphones to enhance the quality of speech

The system uses the array of microphones to locate a speaker in the room, applies the appropriate steering delays to steer the main beam of the array in the direction of the speaker, and sums the outputs of the delayed microphone signals to form the monophonic output signal.

4.1.1. Experimental Procedure

Data were collected in a laboratory room at the CAIP center that was approximately 20 feet by 20 feet, with an 8-ft. ceiling. The walls were built from concrete block and the floor was linoleum tiled. The room was furnished with desks, chairs, lab equipment, and some computers (with noisy fans). Five male speakers were used, each speaking five utterances from an alphanumeric database and speaking answers to nine census questions (also numbers and letters).

Four channels of data were simultaneously collected from each of four different microphones. The microphones were a headset-mounted Sennheiser HMD224 microphone (CLSTK), the omnidirectional Crown PZM6FS table top microphone (CRPZM), and two microphone arrays, one with an 8-kHz bandwidth (as in the standard ARPA evaluations), and one with a 4-kHz bandwidth (telephone bandwidth).

4.1.2. Array System Description

The microphone arrays (Figure 4-1) each had 23 elements, and were actually a combination of three sub-arrays of 11 elements each as discussed in Flanagan *et al.* [1985]. The three sub-arrays each had a different linear spacing of elements, and were bandlimited to one portion of the total frequency range. By limiting the bandwidth to a particular frequency range for a given element spacing, we can avoid spatial aliasing of input signals and keep the beamwidth fairly constant over the entire frequency range of operation of the array. The sub-array with the narrowest spacing (4 cm between elements) covered the highest frequencies (2 kHz - 8 kHz), the sub-array with the intermediate spacing (8 cm between elements) covered the mid frequencies (1 kHz - 2 kHz), and the sub-array with the widest spacing (16 cm between array elements) covered the lowest frequency band (0 - 1 kHz). Some elements could be shared between sub-arrays as the spacing was doubled from sub-array to sub-array. (Figure 4-1 depicts three 7-element sub-arrays, for a total of 15 elements. The actual CAIP Center array has three 11-element sub-arrays, for a total of 23 elements.)

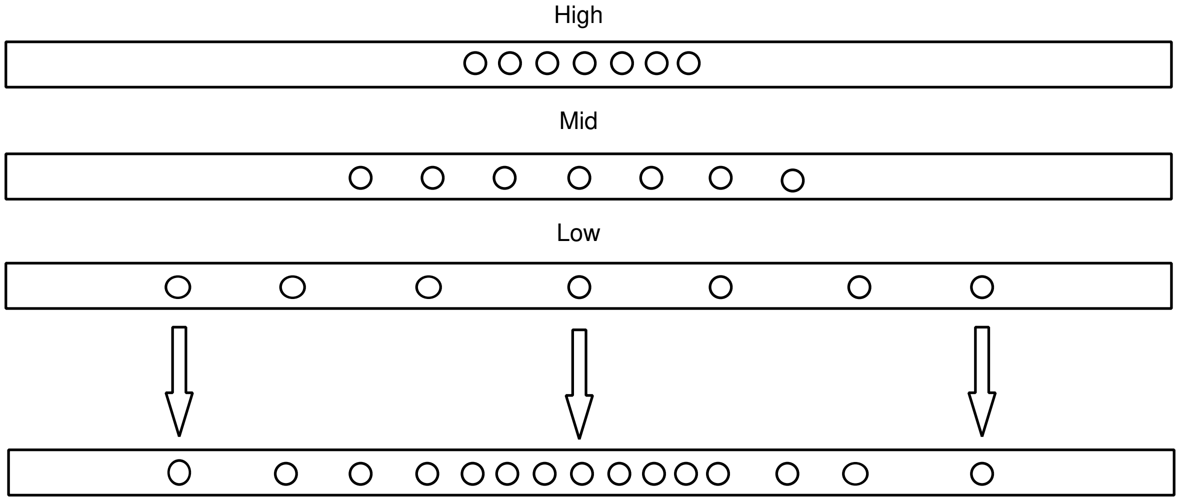

Figure 4-1: CAIP Center delay-and-sum array.

The array elements are inexpensive noise-cancelling electret condenser elements manufactured by Panasonic. The signals from the array elements are delayed prior to summing by analog delay chips in the array hardware. The delays are chosen to set the principal beam of the array in the direction of one of a set of pre-chosen angles from the front of the array. The array is swept through this set of angles to determine the direction of largest energy, and then the delays are frozen to select that direction. Since our subjects were sitting directly in front of the array, the delays were frozen at zero to maximize the array response to the front of the array.

Data were collected with a 16 kHz sampling rate at distances from the microphones of 1, 2, and 3 meters. The headset microphone was used for all trials as well, to provide a control and a baseline number for comparing recognition accuracy.

4.1.3. A Discussion on Spatial Aliasing

The issue of spatial aliasing must be considered when sampling signals in space, much in the same way that frequency and time aliasing are considered when sampling a signal in time. In time, the frequency represented by half of the sampling period is the Nyquist frequency, or the maximum frequency that can be represented by that sampling rate. In space, the maximum frequency for which we can infer azimuth from time delays of signals arriving at two sensors is the frequency with a wavelength of half the distance which causes the arrival time delay between the two sensors. The time delay between two sensors is determined by the spatial distance the signal must travel between the two sensors, and the speed of sound in the propagation medium.

The maximum arrival time delay possible between two sensors would occur when the signal must traverse the entire physical spacing between the two sensors. This occurs when the signal arrives directly from one end of the array, or 90 degrees off axis to the front of the array. The wavelength associated with half the distance between sensors is the wavelength of the worst case, or minimum, spatial aliasing frequency for that array. As the azimuth angle of signal arrival moves closer to the perpendicular bisector to the array, or closer to being on-axis with the array, then the time delay of arrival decreases. This decreases the wavelength of the frequency at which spatial aliasing begins to occur, which therefore increases the value of the frequency itself. For a signal arriving to both sensors at exactly the same time, there is no spatial aliasing for any frequency.

For the array used in this experiment, the frequency bands for each sub-array were chosen to avoid spatial aliasing at any signal arrival angle to that sub-array.

4.1.4. Experimental Results

Figure 4-2 shows the results of our experiments for three of the four microphones: the close-talking headset microphone (CLSTK), the PZM6FS omnidirectional desktop microphone, and the wideband array (because 8 kHz is the bandwidth of the speech collected from the other microphones). Results are shown for the 1- and 3-meter distances. Data are presented with and without CDCN processing for the effects of noise and filtering in the room. The CMU SPHINX-1 speech recognition system was used to test the recognition accuracy for each condition. Results are presented as percentage error rate.

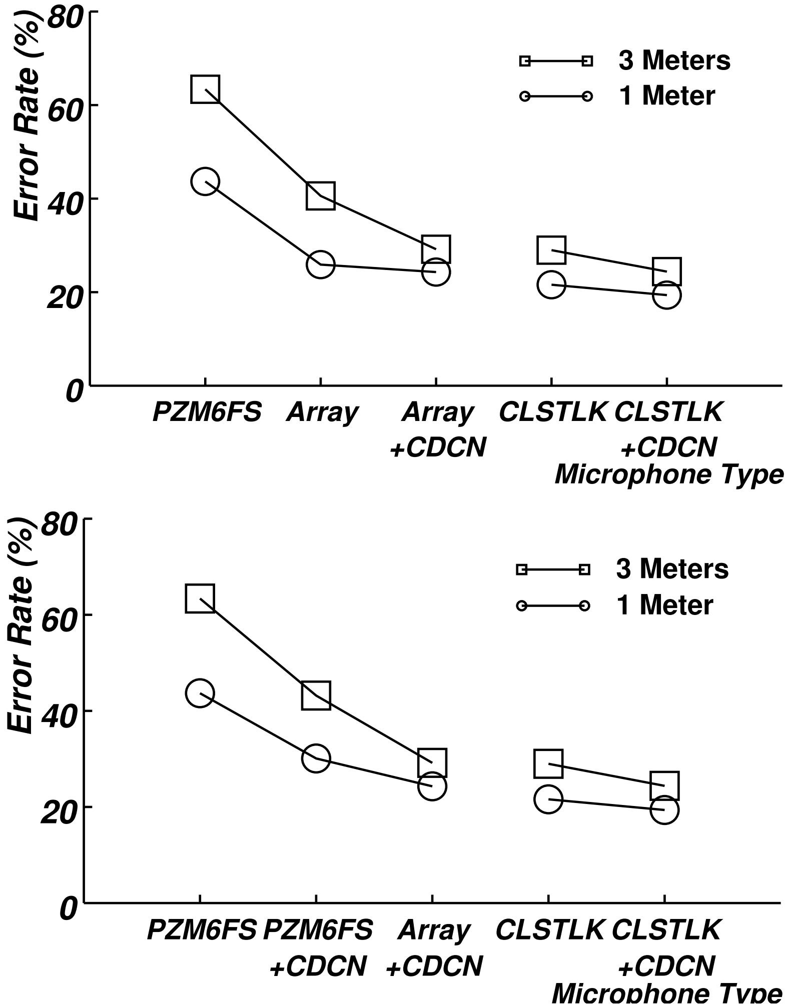

Figure 4-2: Results of pilot experiments performed at the Rutgers CAIP Center.

We observe in Figure 4-2 (upper panel) that without CDCN, the array provides a fair amount of improvement in error rate over the error rates observed using the PZM6FS microphone, at both distances. (Results obtained using the PZM6FS provide the best estimate available to the results that would have been obtained using a single element from the array). We also see that using the array in combination with CDCN provides a further drop in recognition error over both processing the output of the PZM6FS microphone with CDCN (Figure 4-2, lower panel), and over the error rate obtained with the array without CDCN (Figure 4-2, upper panel). The conclusion that we draw from these data is that array processing and environmental normalization algorithms such as CDCN can be mutually beneficial for automatic speech recognition systems. Note also that the performance of the array output processed with CDCN (in all cases) yields a result equal to, or nearly equivalent to, the performance of the close-talking microphone alone.

The difference in recognition error observed using the close-talking microphone at the 1- and 3-meter distances is hypothesized to be a result of small speaker sample size, and possibly overarticulation by the speakers at the three-meter distance (subjects may have felt a need to “shout” or speak with more emphasis the further away from the array that they were).

The results of this pilot experiment convinced us that further research into multi-microphone systems is a worthwhile research direction for speech recognition applications.

Chapter 5. An Algorithm for Correlation-Based Processing of Speech

In this chapter we describe the algorithm with which the majority of the experiments in this thesis were carried out. Our multi-microphone correlation-based algorithm for speech signal processing consists of cross-correlating the rectified output signals from a bank of narrow-band filters. We will first describe the algorithm and then we will discuss the implementation of the various parts of the system. For each of processing sections of the algorithm, we describe the different ways in which that particular leg of the processing was carried out for our experiments. The experiments themselves and the results obtained will be described in detail in Chapter 7.

5.1. The Correlation-Based Array Processing Algorithm

Figure 5-1 shows the block-diagram of the multi-microphone correlation-based signal processing procedure. The signals from the microphones, $x_k[n]$, (where k is the microphone number) are input to a filterbank which roughly models the overlapped filters spaced along the basilar membrane in the human auditory system.

The signals output by each filter are rectified to yield a set of output signals $y_k[n,\omega_c]$ (where c denotes a particular filter band), and then correlated with the filtered and rectified signals in the same frequency band from the other microphones to yield the output “energy” values, E_c.

If there are only two microphones in the array, the correlation equation for the rectified signals would be:

$$E_c = \sum_{n=0}^{N-1} y_1[n,\omega_c] y_2[n,\omega_c]$$

where N is the number of samples per analysis frame, $y_1[n,\omega_c]$ and $y_2[n, \omega_c]$ are the rectified signals from each microphone after filtering by the bandpass filter with center frequency ω_c, and E_c is the “energy” value for that filter channel.

For the general case of K microphones, this equation becomes:

$$E_c = \left\{ \sum_{n=0}^{N-1} \prod_{k=1}^{K} y_k[n, \omega_c] \right\}^{2/K}$$

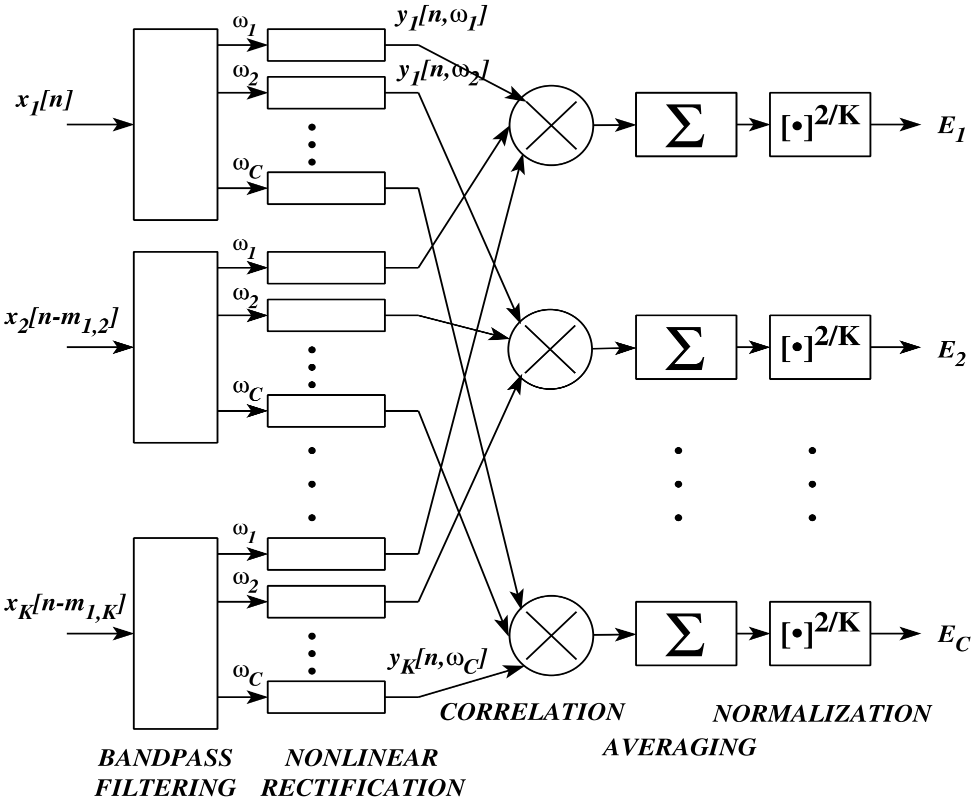

Figure 5-1: Multi-microphone correlation-based signal processing algorithm.

The factor of 2/K in the exponent enables the result to retain the dimension of energy, regardless of the number of microphones. For K=2 this equation reduces to the previous case.

The logarithms of the resulting "energy" values (one for each band of the filterbank) are then converted via the discrete cosine transform (Davis and Mermelstein [1980]) to a set of cepstral coefficients, F_i, to be used as a feature vector for the recognizer (along with an additional frame power value). C is the total number of filterbank channels present in the system and i is the index number for each derived cepstral coefficient.

$$F_i = \sum_{c=1}^{C} (\log E_c) \cos\left[i\left(c - \frac{1}{2}\right)\frac{\pi}{C} \right]$$

5.2. Details of the Correlation-based Processing Algorithm

We now describe each of the processing sections used to implement the algorithm.

5.2.1. Sensors and Spacing

The input sensors used in the arrays for our processing are inexpensive noise-cancelling electret condenser capsules. These microphones are somewhat directional, and therefore reject noise from the rear. They have been found to yield superior performance to omni-directional electret condensers by researchers at AT&T, the CAIP Center, and Brown University, and were recommended to us by these groups. The microphones have pressure-gradient capsules, and as a result have a low-frequency rolloff in their response. After collecting input speech with these microphones, the signals are passed through a digital filter that has a gain boost at low frequencies and unity gain for higher frequencies to flatten the frequency response of these microphones over the 0 - 8 kHz frequency range. From 0 to 125 Hz the filter is flat with a gain of 24 dB; it then drops off at 6 dB/octave from 125 to 1000 Hz and levels off to a 0 dB (unity) gain at 1000 Hz.

The array sensors were plugged into mini-DIN microphone jacks with 10-foot wires. The microphone jacks were pushed from the rear into holes drilled along a strip of wood. The front of the strip of wood was covered with foam rubber, to provide some acoustic dampening of the input speech signals and room noise. The microphone elements were mounted flush with the surface of the foam. The microphone outputs were input to a Model M16 multi-channel A/D converter manufactured by Applied Speech Technologies, Inc. Spacing for the array elements differed from experiment to experiment, and will be discussed in Chapter 7. The M16 multi-channel A/D converter, connected to the DSP port on a NeXT computer, provides simultaneous sampling of up to 16 input channels at a 16-kHz sampling rate. The data are linearly sampled using 16 bits per sample.

For each experiment, a close-talking Sennheiser HMD224 headset microphone was used to collect data concurrently with the microphone array to provide a signal with which to obtain a baseline recognition number. The signal from this microphone was applied directly to the correlation-based processing algorithm.

5.2.2. Steering Delays

Steering delays are applied prior to any other processing to align the array in the direction of the desired signal, which in our case is assumed to be the signal of highest energy incident on the array. With the proper steering delay applied to each microphone, the desired signal will be in-phase at all of the inputs to subsequent processing (in our case, the filterbanks). For the experiments in this thesis, it was assumed that the speaker would remain in the same location for the duration of each utterance. With this assumption, the steering delays would remain constant for the duration of the utterance.

The steering delays are calculated by cross-correlating the signal outputs from adjacent pairs of sensors. The cross-correlation function is searched in a region around the origin to find the location of the peak in the function, and this location is chosen as the sample delay to apply to align the sensors. The delay between each pair of sensors is calculated in the same manner such that the entire array is eventually aligned in the direction of the incoming signal of greatest energy. Only frames in the utterance of high signal energy are used in the steering delay calculation to avoid corrupting the delay calculation by the locations of unwanted noise sources.

Experiments were carried out using the automatically calculated steering delays, without steering delays, and with “hardwired” steering delays. The hardwired delays were determined from the known location of the speaker and the array geometry (which varied across experiments). The signal is assumed to be propagating radially from a point source (*i.e.* in the “near-field” condition), as opposed to arriving as a plane wave (as would a “far-field” signal).

5.2.3. Filterbank

Two bandpass filterbank designs were used for experiments to determine the effect of using different filterbanks. Both filterbanks used a total of 40 filters with center frequencies from 0 to 8 kHz.

The first filterbank was designed by Seneff [1988] at MIT to model the peripheral filtering provided by the human auditory system. The filter is implemented with a cascade/parallel network as shown in Figure 5-2.

- **FILTER A** is an FIR filter with 8 zeros.
- Each of the 40 **FILTER B** units is an FIR filter with two zeros.
- Each of the 40 **FILTER C** units is an IIR filter with 4 poles and 4 zeros.

The composite frequency responses of the 40 filters of the Seneff model are shown in Figure 5-3 (from Ohshima [1993]). The lower-frequency filters have center frequencies that are approximately linearly spaced along the frequency axis, while the center frequencies are more logarithmically spaced at higher frequencies.

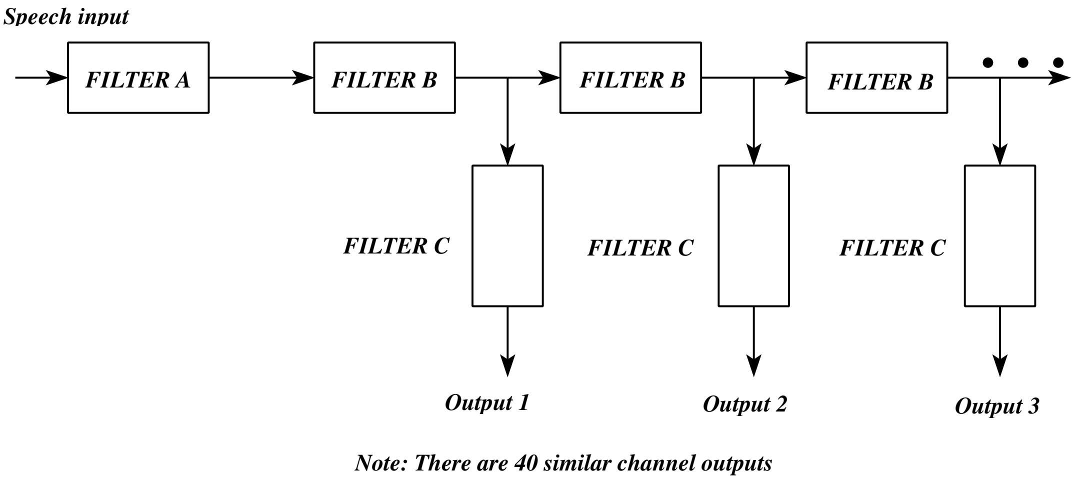

Figure 5-2: Cascade/parallel implementation of the Seneff filterbank.

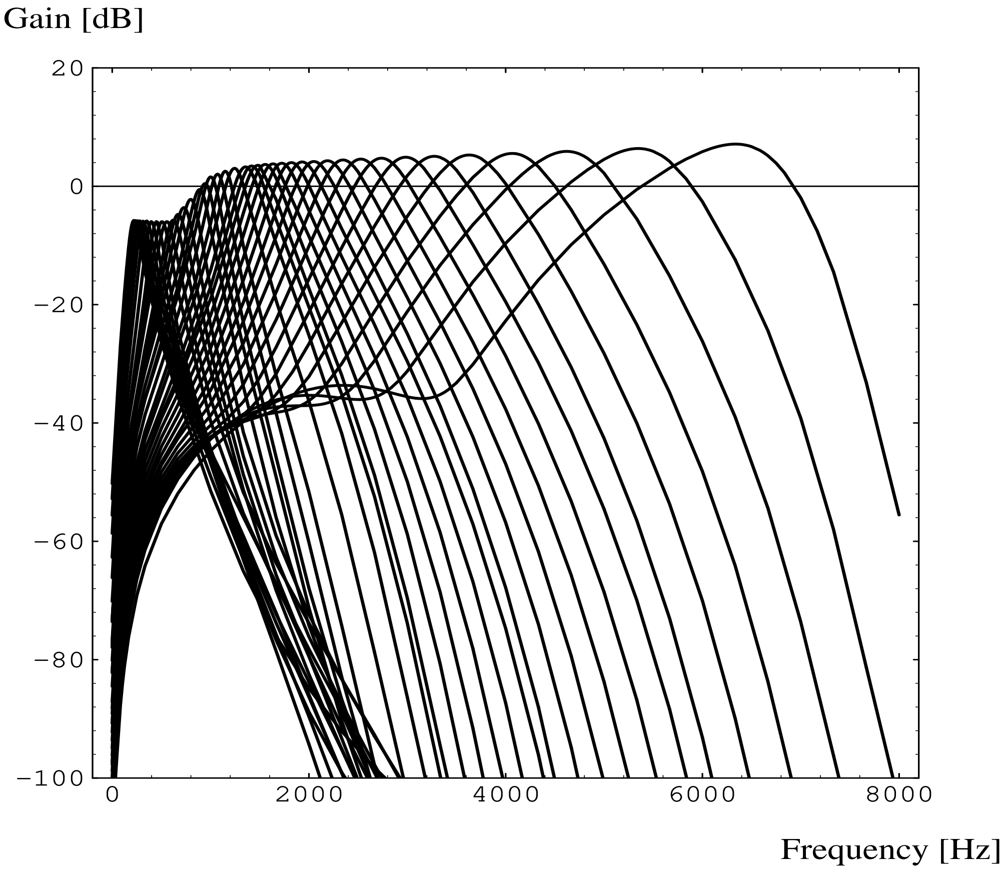

Figure 5-3: Frequency response of the Seneff filterbank.

The second filterbank used in our experiments consisted of mel-frequency filters as described in Davis and Mermelstein [1980]. These filters, have a triangular shape in the frequency domain when plotted on a linear scale, but take on the more rounded "nose cone" shape shown in Figure 5-4 when plotted on a semi-log scale.

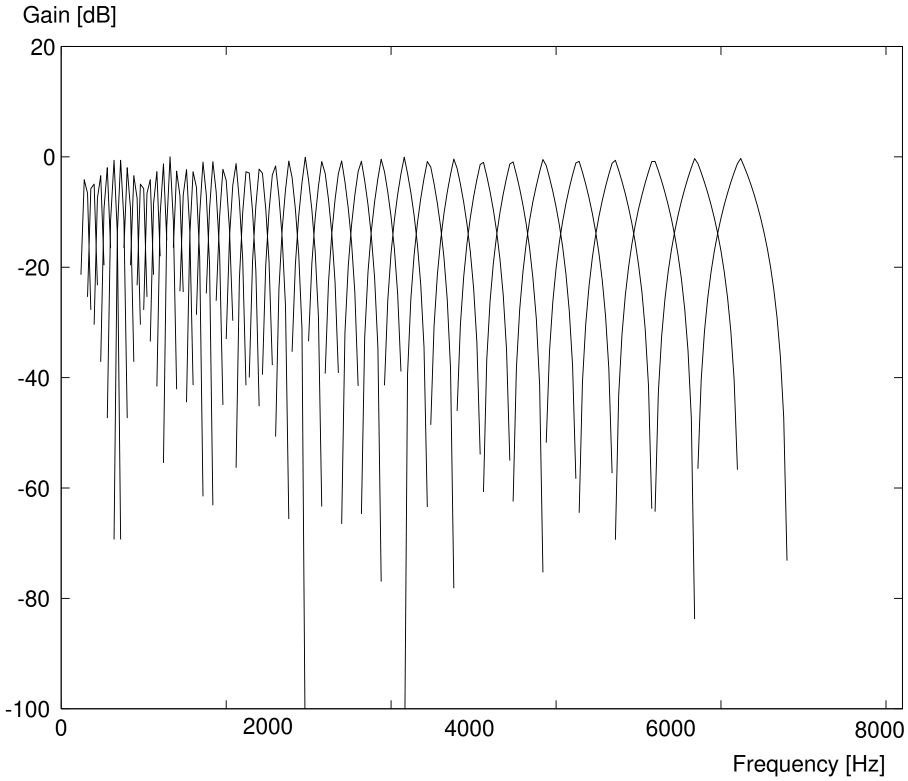

Figure 5-4: Frequency response of the mel-frequency filterbank

The mel-frequency filters are triangular weighting functions for DFT frequency bands. They are linearly spaced with constant bandwidth at the lower end of the frequency range, and logarithmically spaced at the higher frequencies. They are designed by specifying the bandwidth of the linearly-spaced filters (which overlap by half of their bandwidth), the frequency where the center frequencies change from linear spacing to logarithmic spacing, and the total number of filters desired for the entire filterbank.

To apply the mel-frequency filters, a DFT is first taken of the input speech frame. For each mel-frequency filter band, that filter's DFT weighting response is applied to the DFT coefficients of the frame. An IDFT is performed on the result to provide a time-domain signal representing a narrow range of frequencies.

For our experiments, two mel-frequency filterbanks were used. The first had 13 linearly-spaced and 27 logarithmically-spaced filters. The linearly-spaced filters had a bandwidth of 133 Hz and the ratio of the center frequencies of adjacent logarithmically-spaced filters was 1.071. The second filterbank had 15 linearly spaced and 25 logarithmically spaced filters with the linearly-spaced filters having a bandwidth of 100 Hz and the frequency ratio between adjacent logarithmically-spaced filters was 1.085.

5.2.4. Rectification

The non-linear rectification is an important part of our processing. The shape of the rectifier has much to do with what happens when the signals are ultimately combined, because following it is the multiplicative correlation stage. It is our desire to have the rectifier shape favor the desired components of our input signals over those we wish to reject.

We tried a number of rectifiers in our experiments. Using no rectification and using full-wave rectifiers yielded very poor recognition results in initial tests, so they were dismissed from further testing immediately. This was not surprising, as using no rectification or using full-wave rectification leads to frequency doubling and spectral distortion after the correlation operation. The rest of the rectifiers were half-wave in function, having either zero or very small levels for negative portions of the waveform, and passing (with some gain structure) positive portions of the waveform. These half-wave rectifiers preserve the number of positive peaks in the signal, and preserve the original fundamental frequency after correlation.

The Seneff rectifier [Seneff 1988], is modeled on the shape of the rectifiers in our human auditory system. The rectifier is implemented via this equation:

$$y = \begin{cases} G(1 + A\,\mathrm{atan}(Bx)); & (x > 0) \\ Ge^{ABx}; & (x \le 0) \end{cases}$$

For the work in this thesis we used the values: A = 10, B = 0.0004, and G = 450. The negative signals are attenuated in range to a very small level. Even with AB = 0.004, The exponent will be less than -1 for any input value less than -250. The positive signals have a somewhat compressive function applied to them via the arctan function.

The other rectifiers considered belong to the "power-law" series of halfwave rectifiers. The negative portion of the signals is set to zero, and the positive portion of the signal is raised to an integer power greater than or equal to 1. These rectifiers have an expansive response for the positive portions of the input waveforms as shown in Figure 5-5. Our intention in using an expansive rectifier is to provide some weak-signal suppression (or strong-signal emphasis) prior to the correlation operation.

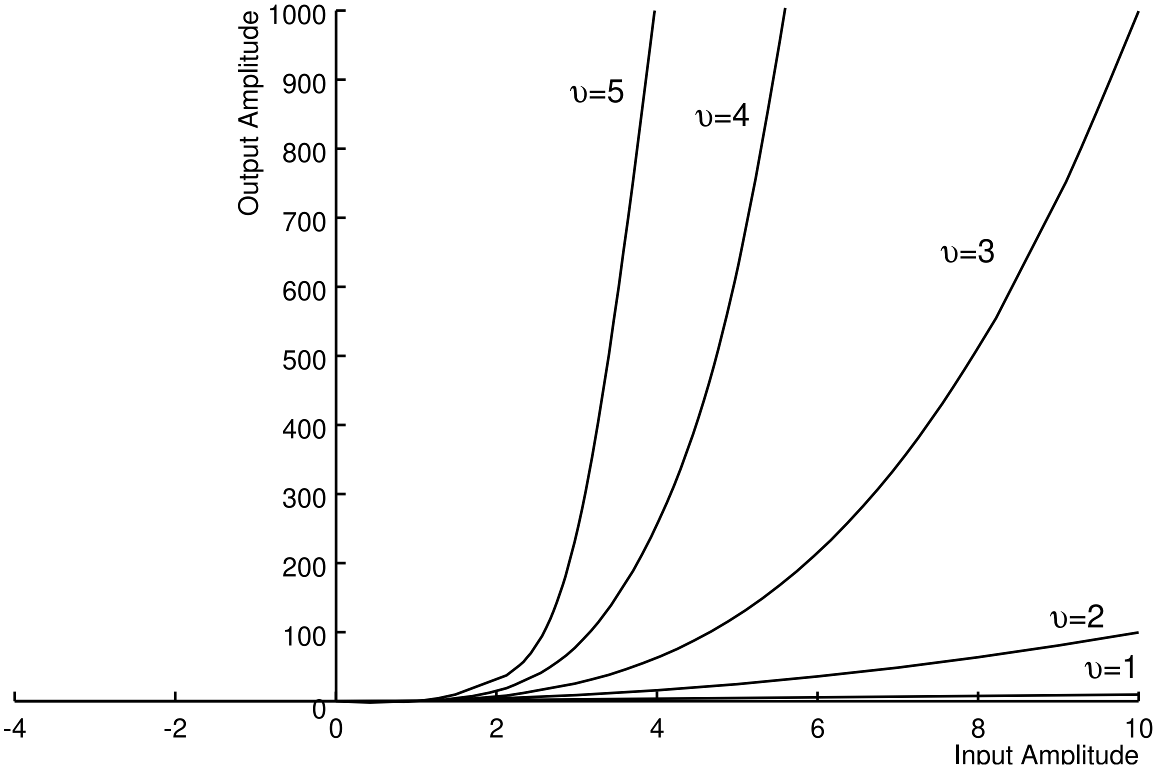

Figure 5-5: Input-output relationships of power-law rectifiers.

5.2.5. Correlation

Since the signals are time-aligned with steering delays prior to entering the initial filterbank in the system, and since any phase or time delays incurred by the signals as they are passed through the various stages of the processing are incurred equally by all sensors, the signals will remain time-aligned as they reach the outputs of the rectifiers. Therefore, the value of the peak of the cross-correlation function will be at the zero time-delay location and we only need to calculate one value of the cross-correlation function for each frequency band. This single value will represent the spectral energy for that frequency band. This frequency band energy value is obtained by multiplying the rectifier output signals of that frequency band from each sensor and summing the product over the length of the frame.

5.2.6. Feature Vector

The feature vector to be presented to the speech recognizer is formed from the 40 spectral energy values output from the correlation operation (one for each frequency band in the filterbank). The natural logarithm of the energy values is first taken, then the discrete cosine transform (DCT) is applied to the 40 log-spectral energy values (Davis and Mermelstein [1985]) to create a set of cepstral-like coefficients. We created a set of 40 cepstral coefficients for some experiments and a set of 12 cepstral coefficients for others. (In test cases, we found the recognition accuracy was not effected by going to the smaller set of 12 coefficients instead of 40 coefficients). The final feature vector is now formed by including an additional feature $c[0]$ that represents the overall frame energy of the frame being processed. This is calculated as the sum of the log-spectral energy values for all of the frequency bands.

5.3. Summary

We have presented an algorithm for processing and combining multiple channels of input speech signals. This algorithm is designed to provide a recognition feature set by using multiple input sensors which provides less recognition error compared to the error obtained from the features derived using a single input (monophonic) sensor.

The next chapter will describe the set of pilot experiments intended to confirm the utility of this algorithm for enhancing speech and to improve speech recognition accuracy.

Chapter 6. Pilot Experiments Using Correlation-Based Processing

Three pilot experiments were performed to verify the hypotheses leading to the design of our cross-correlation based algorithm. The first experiment used sine tones and later narrowband noise passed through simple bandpass filters to represent the peripheral filtering of the human auditory system. The second experiment also used sine and bandpass noise signals, but the input signals were passed through the Seneff filterbank, a more accurate model of peripheral auditory filtering [Seneff,1988]. The third experiment used the vowel /a/ processed through the Seneff auditory filterbank.

The goal of the pilot experiments was to determine the extent to which spectral profiles of incoming signals are preserved by cross-correlation processing in the presence of additive noise sources arriving from a different spatial location.

6.1. Amplitude Ratios of Tones Plus Noise through Basic Bandpass Filters

The first pilot experiment (Figure 6-1) was performed in order to determine if the relationship between desired spectral components would be preserved by cross-correlation type processing and if they could be recovered from signals in the presence of additive uncorrelated noise. We examined only the amplitude ratios of two sine waves in additive noise after the filtering, rectification, and correlation stages, modelling the initial processing in the human auditory system.

More specifically, two sine tones ($s_1(t)$ at 1 kHz and $s_2(t)$ at 500 Hz) with an amplitude signal ratio (A_1/A_2) of 40 dB were corrupted by additive white Gaussian noise. The amplitude of the noise was varied to manipulate the SNR of the signals. The signals were passed through two bandpass filters, with the same center frequencies as the two sine tones. The bandpass filters were FIR filters designed using the Parks-McClellan algorithm [Parks and McClellan, 1972]. $H_1(f)$ had a center frequency of 1 kHz and a bandwidth of 100 Hz; $H_2(f)$ had a center frequency of 500 Hz and a bandwidth of 50 Hz. The output signals were rectified using a half-wave linear rectifier. The noise was added with a time delay T_d between each successive input sensor, which simulates the delay that would be experienced by the noise source if it had arrived at a linear microphone array from an oblique angle. The outputs of the rectifiers at each of the two frequency bands were separately correlated, and the output energy, E_1 and E_2, was then compared across the two bands.

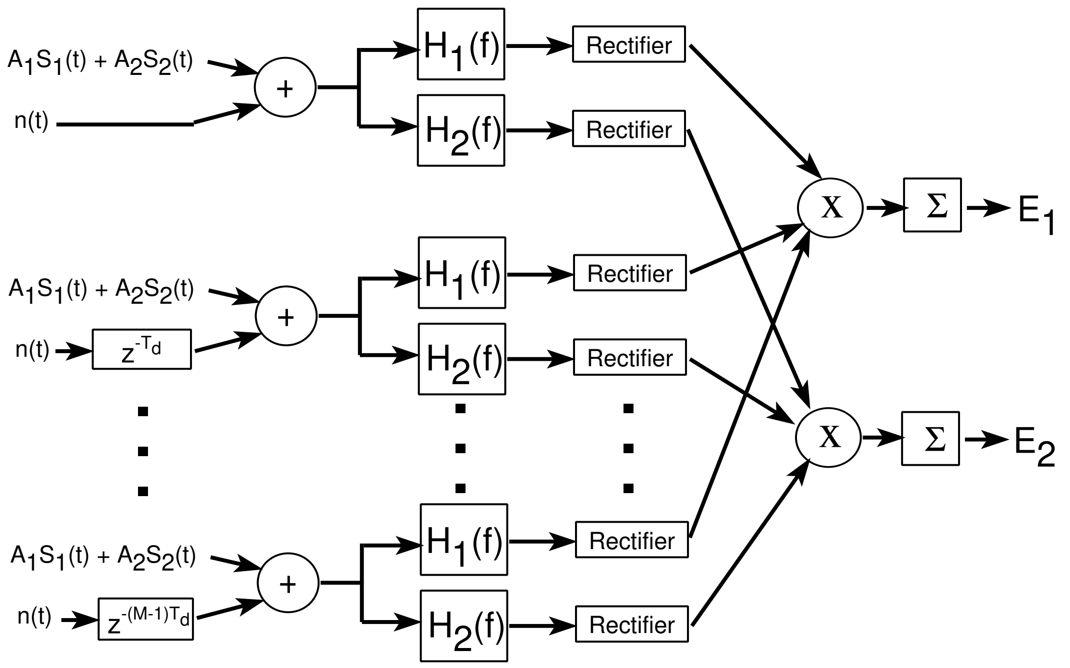

Figure 6-1: Block diagram of the stimuli and processing stages of Pilot Experiments #1 and #2.

Data were obtained by observing the output energy ratio (E_1/E_2) of the signals arriving from the 1-kHz and 500-Hz channels as a function of the SNR of the more intense 1-kHz tone and the additive noise. The SNR was varied from 0 dB to 30 dB. This was done for a variety of time delays of the noise signals running from 0.0 to 1.0 ms. The number of microphones (signal inputs) was also varied. We show the results for 2 and 8 microphones to observe the effect of the processing as more input microphones are used. The noise time delay was constant between adjacent input microphones. What we desire in this pilot experiment is to recover the original input energy ratio E_1/E_2 of 40 dB in conditions of low SNR.

Figures 6-2 and 6-3 show plots of the output data obtained for the 2- and 8-microphone cases. The curves represent the energy ratio E_1/E_2 in dB at the outputs of the bandpass filters centered at the input tone frequencies versus the SNR (between the 1-kHz input tone and the additive Gaussian noise) at various noise time delays T_d between adjacent microphones. In both Figure 6-2 and Figure 6-3; for curve (a)the lowest curve represents a noise time delay of 0 ms (noise arrives on-axis), and each successive higher curve is an additional delay of 0.0625 ms (0 ms, 0.0625 ms, 0.125 ms, 0.1875 ms, and 0.25 ms) and for curve (b) the lowest curve represents a noise time delay of 0 ms (noise arrives on-axis), and each successive higher curve is an additional delay of 0.25 ms (0 ms, 0.25 ms, 0.50 ms, 0.75 ms, and 1.00 ms).

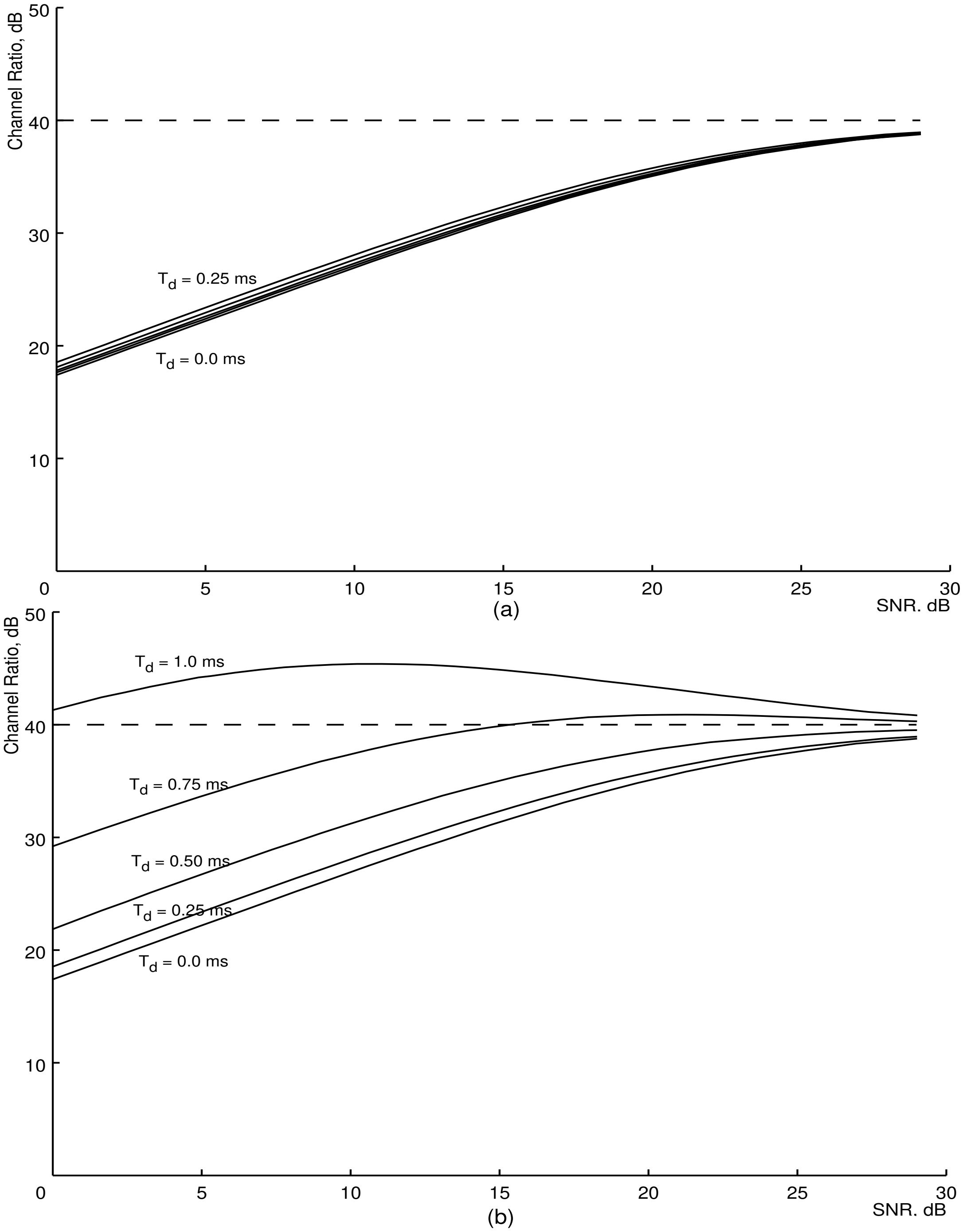

Figure 6-2: Amplitude ratio of a 2-tone signal in noise using cross-correlation processing with 2 microphones. Panel (a) shows predictions using delays from 0 to 0.25 ms in 0.0625-ms steps, and panel (b) shows predictions for delays from 0 to 1 ms in 0.25-ms steps.

In Figure 6-2 and Figure 6-3 we see that at zero time delay the energy ratio emerging from the channels is very different from the "correct" ratio of 40 dB indicated by the dotted lines. This is a result of distortion introduced by the additive noise.

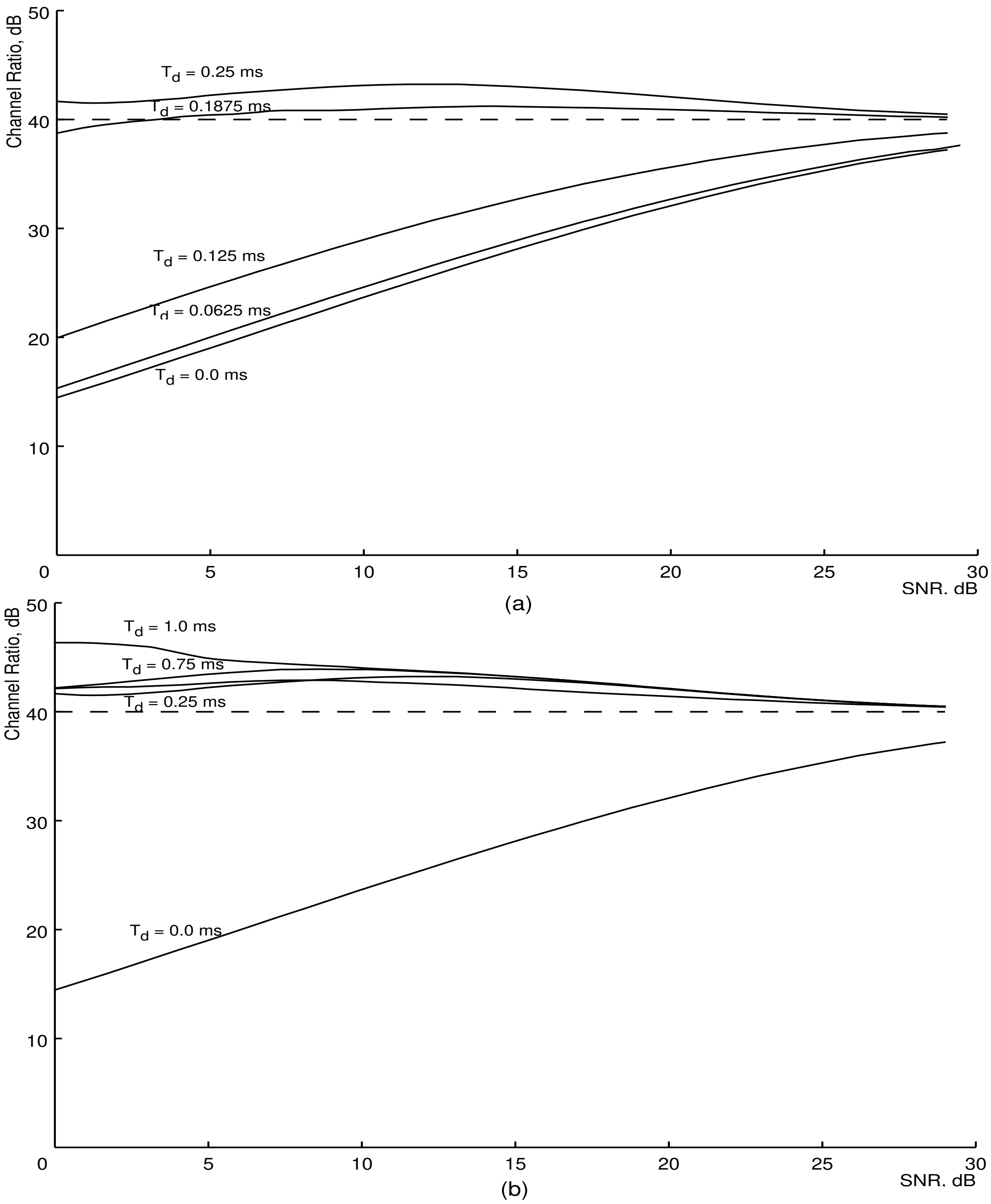

Figure 6-3: Amplitude ratio of a 2-tone signal in noise using cross-correlation processing with 8 microphones. Panel (a) shows predictions using delays from 0 to 0.25 ms in 0.0625-ms steps, and panel (b) shows predictions for delays from 0 to 1 ms in 0.25-ms steps.

As the time delay is increased, we see that the channel ratio becomes closer to 40 dB at the low SNR values. This shows that as the noise source is moved off-axis, the correlation-based system rejects the noise portion of the signal. As the delayed time of arrival of the noise to the microphones increases, it moves the contribution due to the additive noise away from the zero delay position in the cross-correlation plot, whereas the contribution from the desired signals remains at the zero delay position. (Recall from Section 5.2.5 that we only need to look at the zero delay value of the cross-correlation plot because the sensor signals are steered in the direction of the desired signal prior to any of the correlation-based processing.)

As the number of microphones is increased to 8 (Figure 6-3) we see that at any given time delay, the robustness to the noise becomes much improved compared to the two-microphone case due to a sharpening of the cross-correlation curve. For that matter, results for all noise time delays from 0.25 ms to 1.00 ms are nearly indistinguishable in Figure 6-3. This demonstrates that as the number of microphones is increased, the correlation-based processing can reject additive Gaussian noise signals with smaller spatial separations relative to the arrival direction of the desired signal than an implementation using a lower number of microphones.

Figure 6-4 describes the 2-microphone results obtained by summing the signals after bandpass filtering from the respective microphones (as opposed to correlating them), and Figure 6-5 shows the corresponding 8-microphone results. This is comparable to what delay-and-sum processing would produce for these stimuli. We see that as the sensor-to-sensor noise time delay T_d is increased, the amplitude ratio gain for the low SNR cases is increasing, but by much less than that of the 2-microphone correlation-based case (Figure 6-2). Note that the best we do for the 2-microphone case at any non-zero time delay is a 3-dB increase over the zero time delay case. The best attainable SNR gain for delay-and-sum systems is a 3-dB gain for every doubling of the number of microphones (if the noise is additive white Gaussian noise which is uncorrelated with the signal). We also observed this between the extremes of time delays (0 ms, which represents no time delay, and 1 ms, which represents one period of the 1-kHz tone). Thus, the correlation-based processing appears to provide greater robustness to additive Gaussian noise than does simple delay-and-sum processing.

We repeated this pilot experiment using narrowband noise signals instead of the pure sine signals. The narrowband noise components were 30 Hz in bandwidth and centered at 1 kHz and 500

Hz, respectively, with an amplitude ratio of 40 dB. We observed similar results in recovering the amplitude ratio of the narrowband noise signals as we did for the purely sinusoidal signals.

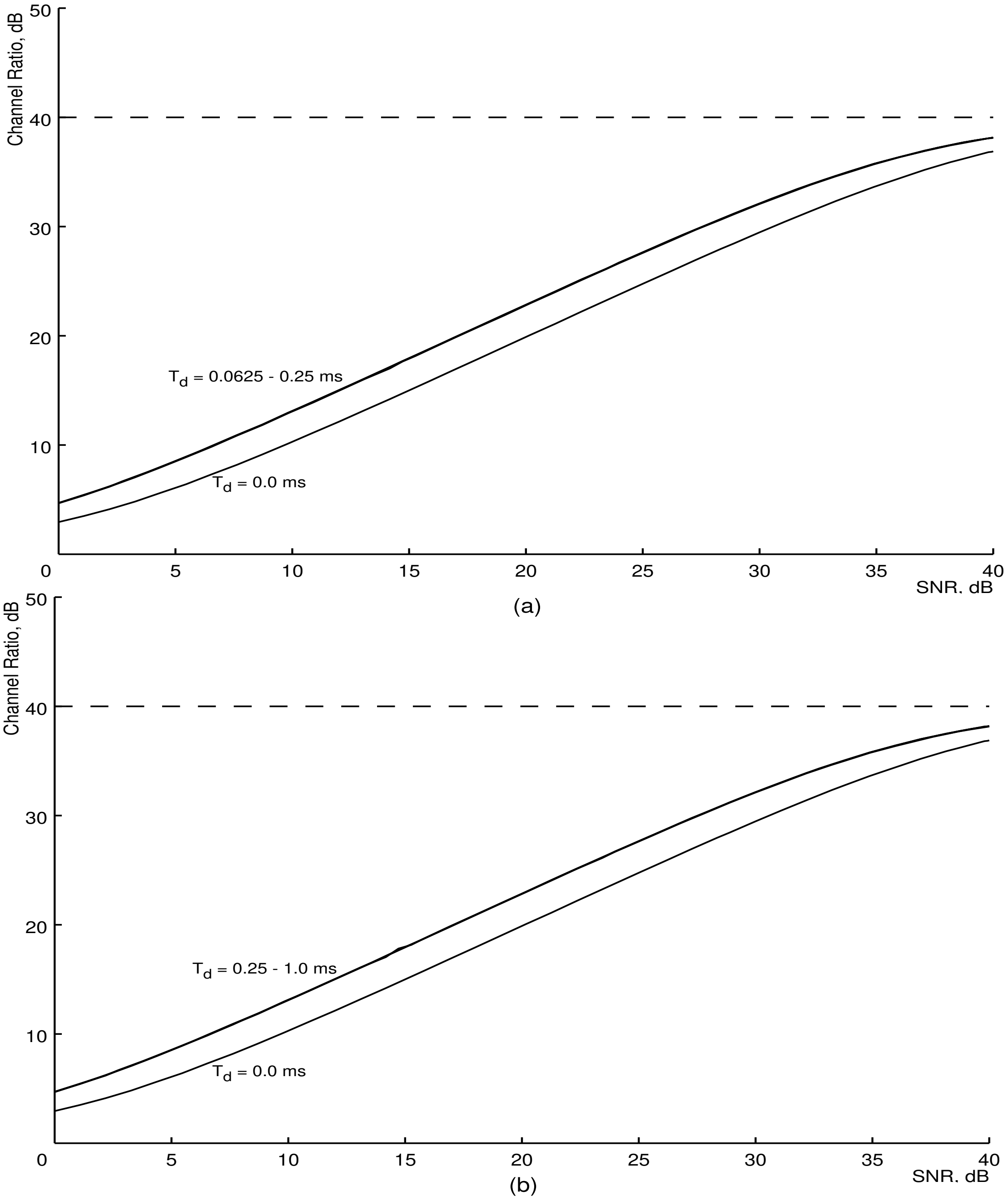

Figure 6-4: Amplitude ratio of a 2-tone signal in noise using delay-and-sum processing with 2 microphones. Panel (a) shows predictions using delays from 0 to 0.25 ms in 0.0625-ms steps, and panel (b) shows predictions using delays from 0 to 1 ms in 0.25-ms steps.

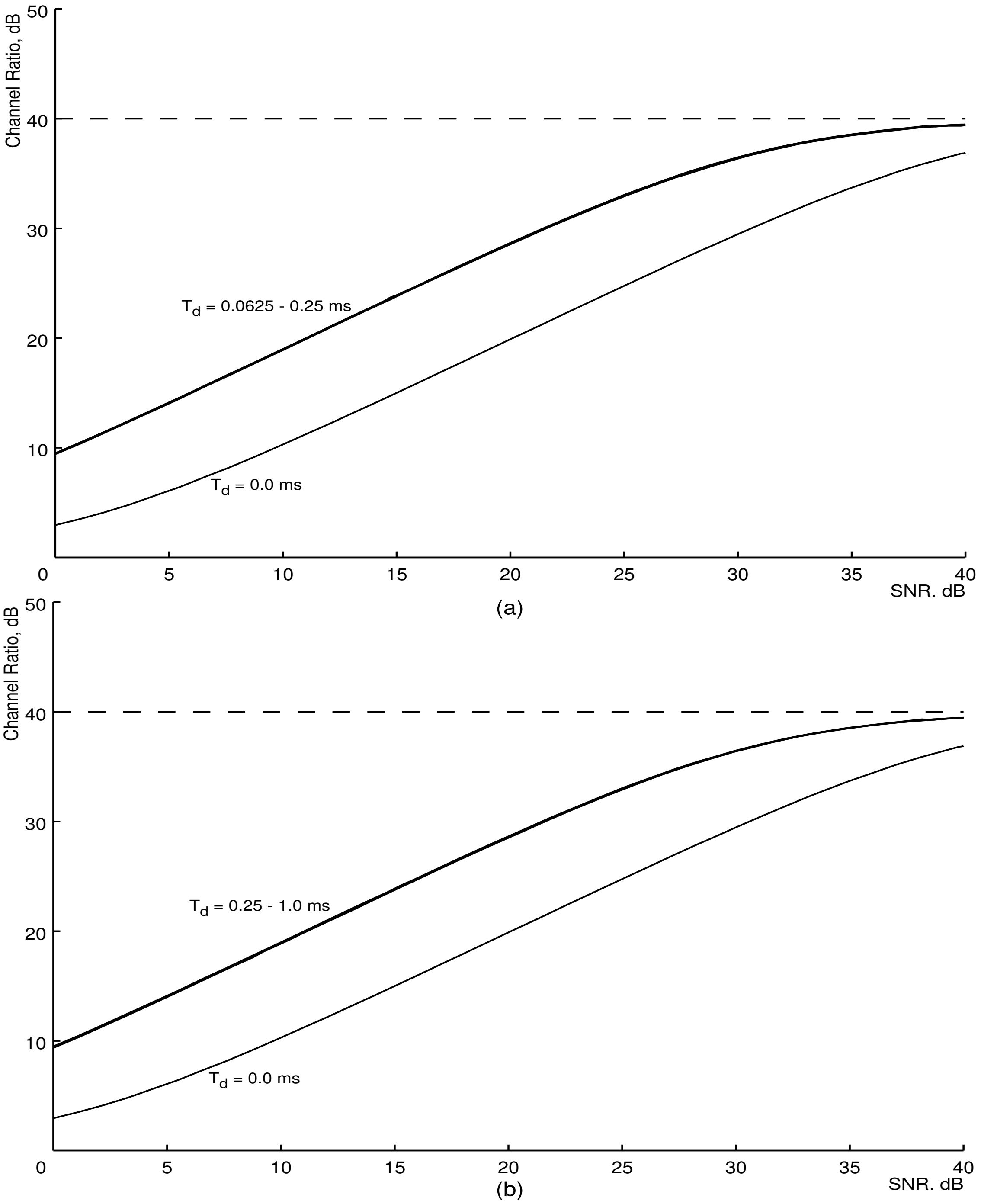

Figure 6-5: Amplitude ratio of a 2-tone signal in noise using delay-and-sum processing with 8 microphones. Panel (a) shows predictions using delays from 0 to 0.25 ms in 0.0625-ms steps, and panel (b) shows predictions using delays from 0 to 1 ms in 0.25-ms steps.

6.2. Amplitude Ratios of Tones Plus Noise through Auditory Filters

The second pilot experiment made use of the same two summed sine wave input signals and delayed additive white noise signals as the first experiment, but these signals were passed through the Seneff filterbank [Seneff, 1988] instead of the basic FIR bandpass filters used in Section 6.1. The outputs of the two bandpass filters used from the Seneff filterbank were centered at 1 kHz and 500 Hz. They were passed through a rectification stage, and we then applied the correlation-based processing. The rectifier used was the non-linear rectifier designed by Seneff for her auditory model (see Section 5.2.3), which is designed to implement the amplitude compression that takes place in the processing in the human ear. The rectifier uses a scaled arctan function if the input signal is greater than zero, and a decaying exponential function if the signal is less than zero.

The shapes of the output curves obtained from each case were similar to the ones in the previous pilot experiment, demonstrating the same robustness to the noise as the previous experiment. This suggests that the exact shape of the bandpass filters from the filterbank used may not be as crucial as the fact that frequency separation via a filterbank is being performed in the first place.

6.3. Spectral Profiles of Speech Corrupted with Noise

Our final pilot experiment used an actual speech signal, a segment of the vowel /a/, as the input to the correlation-based processing system. The signal was obtained from a speech segment collected using a close-talking microphone with a high SNR. The signal was corrupted by adding white Gaussian noise to it prior to its introduction to the Seneff filterbank. The noise was time delayed to each successive microphone input as in the previous pilot experiments. The SNR was varied in 1-dB steps from 0 to 21 dB. The rectifier used was the Seneff rectifier.

Spectral profiles of the vowel /a/ were obtained by plotting the individual frequency band energy values which are output by the correlation-based front end processing. These are shown in Figure 6-6. Figure 6-6a shows profiles for the two-microphone case with zero time delay between microphones from the noise (*i.e.* the noise arrives from the same direction as the signal). Figure 6-6b shows the 2-microphone case with a 125-µs delay of the noise between adjacent input microphones, and Figure 6-6c shows 8 input microphones and 125 µs delay of the noise between adjacent microphones.

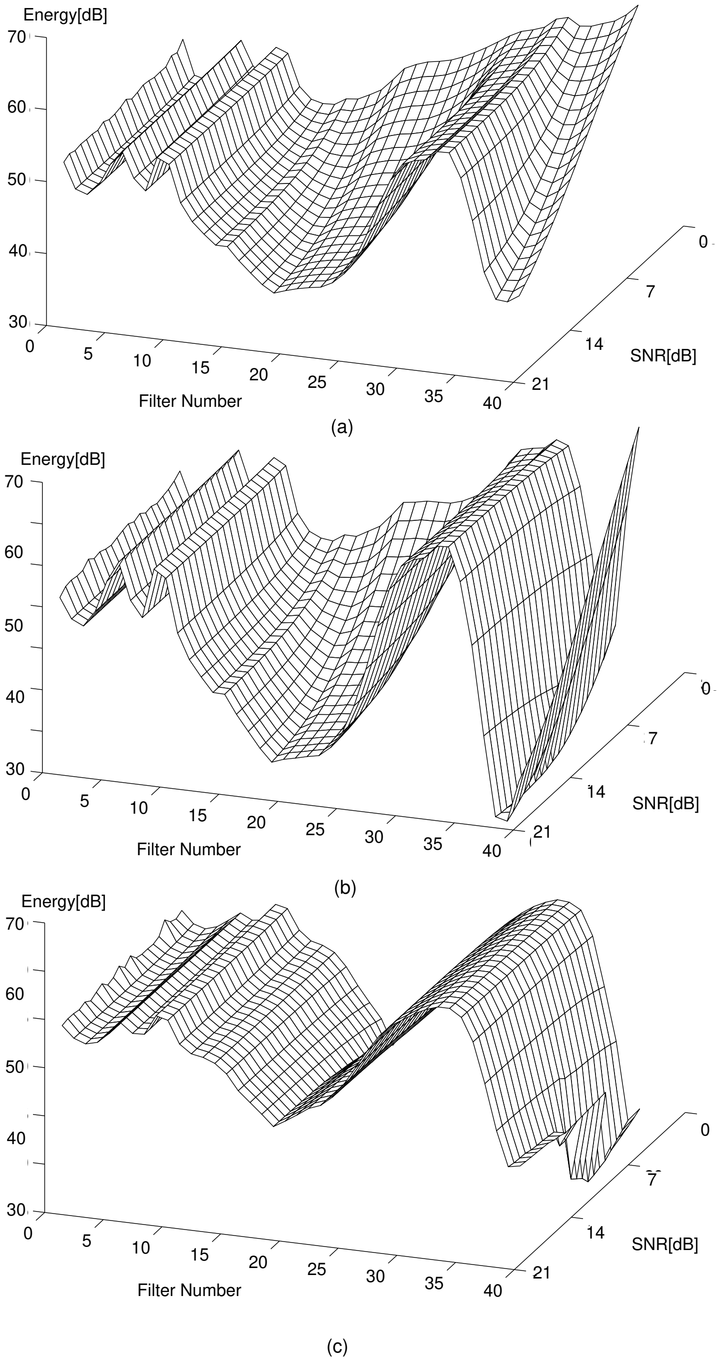

Figure 6-6: Spectral profiles of the vowel /a/ in the presence of noise using cross-correlation processing, for 2 microphones, no delay [panel (a)], 2 microphones, 125-μs delay [panel (b)], and 8 microphones, 125- μs delay [panel (c)].

Figure 6-7 shows the same set of plots using the delay-and-sum processing.

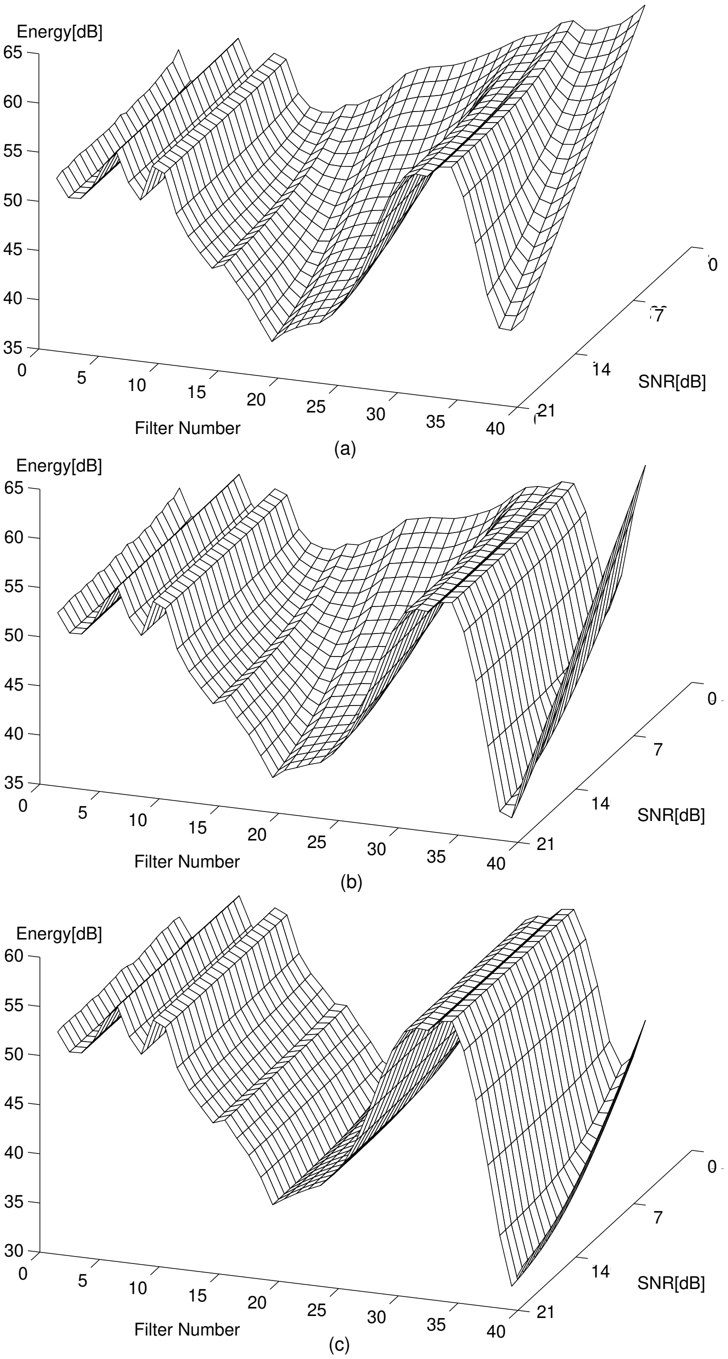

Figure 6-7: Spectral profiles of the vowel /a/ in the presence of noise using delay-and-sum processing, for 2 microphones, no delay [panel (a)], 2 microphones, 125-μs delay [panel (b)], and 8 microphones, 125- μs delay [panel (c)].

Note that in the two-microphone case with zero noise time delay (Figure 6-6a) how corrupted the vowel spectrum is at the lowest SNR (0 dB) and how defined it becomes at the highest SNR (21 dB). This is to be expected, as the on-axis noise will not corrupt the resulting spectrum much if the desired signal is much stronger, but corrupts it greatly as the noise strength increases. As we delay the noise to the other microphone by 125 µs, which corresponds to a two-sample delay for speech sampled at 16 kHz (Figure 6-6b), we see a spectrum at the lowest SNR (0 dB) that is somewhat more equal to the one at 21 dB, and certainly more defined than the 0-dB spectrum in Figure 6-6a. If we next observe the plot using eight microphones (Figure 6-6c), the 0-dB SNR spectrum is almost identical to the 21-dB SNR spectrum. Thus, it appears that the correlation processing provides robustness for actual speech signals in the presence of off-axis additive Gaussian noise. And, as the number of microphones increases, the system is more robust (even in low-SNR conditions) to signals corrupted by slightly off-axis noise sources.

The same trend is observed with the delay-and-sum processing (Figure 6-7). In the zero time-delay case (Figure 6-7a) the plot is very similar to the cross-correlation plot of Figure 6-6a. This is to be expected, as the same signal is being input to all of the sensors. In the 2-microphone case where the noise is delayed between microphones by 125 µs (Figures 6-6b and 6-7b), the cross-correlation processing seems to be recovering the 0 dB signal a bit better than delay-and-sum though delay-and-sum is still providing some improvement. As we go to the 8-microphone case (Figures 6-6c and 6-7c), we see that both delay-and-sum and cross-correlation processing provide a very good improvement in the spectral shape.

6.4. Summary

In summary, these pilot experiments argue convincingly that the correlation-based processing we are researching should be very useful for signal enhancement for speech recognition. Most notable are the results of the experiments when the number of input microphones is increased. In these cases, the system becomes more robust to input noise sources with smaller time delays between the input microphones (corresponding to noise components with smaller spatial separations from the desired signal).

We have also demonstrated that a correlation-based system provided a somewhat greater performance in these test conditions to processing with a delay-and-sum system. In the first two experiments, where pairs of tones were used as the desired signals, we found that the correlation-based processing substantially out-performed the delay-and-sum processing when the jamming noise source was arriving with a large time delay to adjacent sensors, but the performance gain was somewhat lessened as the jamming signal delay time was decreased. The difference was not as great in the third experiment where a wideband signal (the vowel /a/) was the desired signal and the noise arrived at a smaller delay time between adjacent sensors. We do believe the correlation-based processing can provide better performance in actual speech recognition experiments compared to delay-and-sum and traditional MMSE algorithms in situations where the environment is corrupted by linear filtering and reflections of the input speech. Chapter 7 will present the results of actual speech recognition experiments using the algorithm.

Chapter 7. Speech Recognition Experiments

In this chapter we present the results of a set of experiments that were performed to test the various processing blocks of our correlation-based processing algorithm. Experiments were also carried out to compare the performance of the correlation-based array system to that of delay-and-sum beamforming and Griffith-Jim beamforming, a traditional LMS array processing algorithm.

Each of the experiments performed involved running a speech recognition task using the CMU SPHINX-I system. As noted in Chapter 3, the SPHINX-I system used in all of these experiments uses discrete HMMs [Lee, 1988]. There have been numerous improvements to speech recognition technology since this research was begun. For example, SPHINX-II, which uses semi-continuous HMMs [Huang, 1993], has been the standard CMU recognition system for a number of years now, and SPHINX-III, which uses fully continuous HMMs, is now undergoing initial testing. Nevertheless, we elected to continue our experiments with the older SPHINX-I system to provide some consistency of results over the duration of this work. We believe that the gain in recognition accuracy obtained using our correlation-based processing is independent of the type of recognizer used. This is due to the fact that all of our processing is involved in the generation of the feature set, and therefore takes place prior to the application of the feature set to the speech recognizer. Any results obtained using our cross-correlation-based processing should translate equally well to a better-performing recognizer.

The goal of the initial series of experiments was a better understanding of how specific design choices and parameter values affected the recognition accuracy obtained with the cross-correlation-based system. These experiments considered the effects of rectifier shape, the number of input channels, the specific ways in which the steering delays were implemented, and the spacing of the microphones. During the course of these experiments it became clear that the recognition accuracy obtained using the correlation-based system was not as much better than that obtained with simple delay-and-sum beamforming as we had expected on the basis of the results of the pilot experiments described in Chapter 6. We initiated a second set of experiments that were intended to provide a deeper understanding of the correlation-based processing in comparison to delay-and-sum beamforming, along with a smaller number of experiments that compared both methods to the traditional Griffiths-Jim algorithm.

Of necessity we used several different recording environments and array configurations for the various experiments described. Initial experiments were carried out using data recorded in a room with a relatively high ambient noise level called the “computer laboratory”. This room was approximately 10 feet by 20 feet and contained many computer fans, disk drives, etc. The floor was carpeted with very flat indoor/outdoor institutional carpeting and the walls were painted sheet rock plasterboard. The ceiling is about 10 feet high and has a rough mildly absorbent material coating it. The background noise in the room was approximately 70 dB ‘A’ weighted, and typical SNRs of speech recorded in this environment using a non-closetalking microphone were on the order of 4 to 7 dB. These SNRs (and all SNRs reported in this chapter) were calculated using the package developed by the National Institute of Standards and Technology (NIST) [Pallett *et al.*, 1996].

Some of the subsequent experiments were carried out using data in a conference room, which exhibited a higher SNR. The conference room was approximately 20 ft. long by 12 ft. wide, with a 10-ft.-high ceiling. Wall-to-wall institutional carpeting was on the floor with painted sheet rock walls. A 5 ft. by 10 ft. table sits in the middle of the room. There were no computer fans in the room except for the fan in the computer used for data collection. Typical speech recorded in this environment using a non-closetalking microphone is on the order of 14 to 18 dB.

Finally, a smaller number of speech recognition experiments were conducted using data that were obtained by artificially contaminating clean speech with additive noise. Similarly, the choices of microphones, array configuration, and recording procedures evolved as our experimental needs changed and as the available equipment resources became greater.

7.1. Effects of Component Choices and Parameter Values on Recognition Accuracy

7.1.1. Effect of Rectifier Shape

The first speech recognition experiments were performed to examine the effects of varying the type of rectifier used on the filter outputs. The rectifier providing the best performance would be used consistently throughout the experiments examining other parts of our signal processing.

7.1.1.1. Experimental Procedure

The training microphone we used was the CRPZM microphone and the database was that described in Section 3.5. The testing database consisted of 138 different alphanumeric and census utterances from 10 male speakers recorded in stereo using a matched pair of CRPZM microphones in the Computer Laboratory. The speakers were seated in a chair located approximately one meter directly in front of the microphone pair (on axis). The microphones were underneath the computer monitor. The utterances were sampled at a rate of 16 kHz in 16-bit linear signed format. The microphones were spaced approximately 7 cm apart, a physical limitation on the minimum spacing size due to the size of the microphone base. At a 7 cm spacing, frequencies for signals above (approximately) 2450 Hz may be subject to spatial aliasing depending on the direction of their arrival. This was not an issue for the direct speech, but reflected signals and noise sources arriving off-axis may be affected by spatial aliasing.

The testing data were run through the correlation-based processing with each microphone signal processed independently through the filterbank and rectifier sections. The filterbank used was the Seneff filterbank described in Section5.2.3. The rectifier outputs from each microphone were correlated, and “energy” values, cepstral coefficients, and feature vectors were produced as described in Section 5.2.6. The testing data used the same values of frame width and overlap as the training data.

Recognition results were obtained for the testing database using the left and right microphones treated as monaural signals, and using the stereo pair. The monaural analyses enabled comparisons of the recognition results using two microphone inputs with those of a single microphone.

We trained and tested the recognition system using several different rectifiers to observe the effect of rectification on recognition accuracy. The following rectifiers were used:

- **No rectifier:** The unrectified signal is passed on without change.
- **Seneff Rectifier:** As described in Chapter 5, this rectifier has an inverse tangent response for positive signal values and a decaying exponential response for negative signal values. It compresses the dynamic range of the positive portion of the signal.
- **Fullwave Rectifier:** The output of this rectifier is the absolute value of the input.

- **ν-law Rectifiers:** This class of rectifiers raises positive signal values to the ν^{th} power, and sets negative signal values to zero. Therefore, ν=1 is a halfwave rectifier, ν=2 is a halfwave squaring rectifier, etc. These rectifiers are expansive for positive signal values, for values of ν that are greater than 1 (see Figure 5-5).

7.1.1.2. Experimental Results

Figure 7-1 shows the word recognition accuracy obtained for the various rectifiers considered. The lower curve (square symbols) shows the averaged word accuracy obtained using monophonic processing and the upper curve (circular symbols) shows the word accuracy obtained for the stereo case. These and other word accuracy statistics are corrected for insertion penalties.

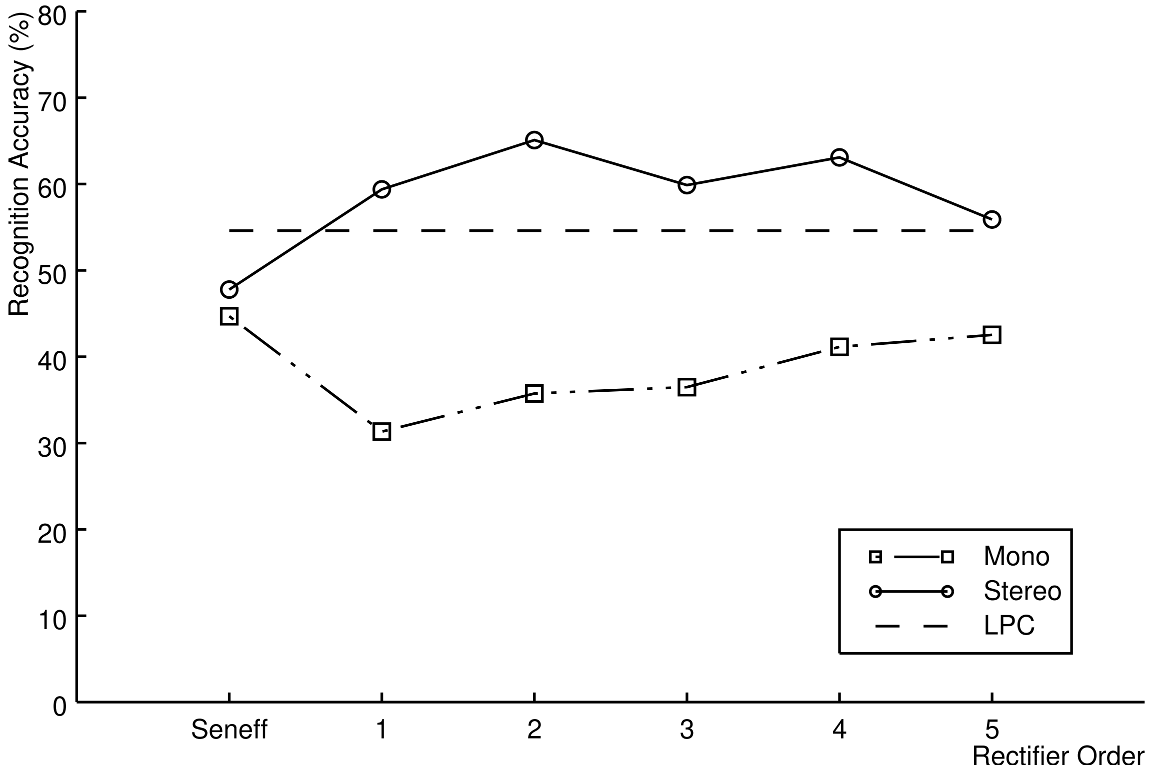

Figure 7-1: Effect of rectifier type on recognition accuracy using correlation-based processing.

The dashed curve in Figure 7-1 without symbols indicates baseline results for the same testing and training databases using the standard monophonic SPHINX-I LPC-based signal processing and feature extraction algorithms. The feature vectors used for these experiments consist of 40 cepstral-like coefficients derived from the 40 energy values obtained from each frequency band per 20-ms frame of windowed speech. There is also one extra cepstral coefficient per frame which represents the average signal power of that frame. For the LPC case, 12 cepstral coefficients represent each frame of data with an additional coefficient that represents the power of each frame.

We note that some sort of half-wave rectification is necessary for the correlation-based processing. When no rectifier is used on the stereo input task, the word recognition accuracy corrected for insertion penalties is -2.2%. This is a result of the squaring nature of the correlation operation. In the no-rectifier and the full-wave rectifier cases, the squaring operation will always result in a positive value output from the correlation operation, but the frequency of the correlation output is double the frequency of the original signal. This leads to a high degree of spectral distortion.

In addition, with full-wave rectification the correlation of two "signal-plus-noise" input terms produces a squared signal term (which is positive in sign), a squared noise term (also positive), and a cross term from the noise-signal product. If the SNR is large, the squared signal term will be large compared to the cross term and the squared noise term. If the signal and noise levels are roughly equivalent, the cross term will be large and may contain negative values. Either situation will produce spectral profiles with a great deal of harmonic distortion.

As can be seen Figure 7-1, the performance of the Seneff rectifier in the context of the correlation-based processing is much worse than that of any of the halfwave ν-law rectifiers. We believe that this occurs because the Seneff rectifier is compressive. We obtain a very small gain in recognition accuracy using this rectifier when replacing a mono input with stereo inputs.

The best results were obtained using the ν-law rectifiers, which all provided better performance than baseline monophonic LPC cepstral processing. In general we believe that the ν-law rectifiers are successful because they are expansive for positive signals for $\nu > 1$. The best recognition accuracy was obtained in the $\nu = 2$ (square-law) case. We hypothesize that this is due to the addition of some gain expansion, but not too high a degree of expansiveness. Since the square-law rectifier provided the best accuracy (by a small margin), we will continue to use it as the default rectifier for the remaining experiments in this chapter. We do not understand exactly why recognition accuracy seems to vary so much for rectifier type used for the monophonic signals.

7.1.2. Effect of the Number of Input Channels

Encouraged by the observation that two microphones provides better recognition accuracy than a single microphone, we increased the number of sensors, comparing recognition accuracy obtained using 1-, 2-, 4-, and 8-element linear microphone arrays.

7.1.2.1. Experimental Procedure

In this and later experiments, speech recognition accuracy obtained using a linear microphone array is compared with the accuracy obtained using the conventional close-talking and the pair of CRPZM omnidirectional desktop microphones described previously.

The microphone array had up to 8 elements, with a 7-cm spacing between adjacent elements. This spacing was chosen because it was equal to the minimum possible spacing for the two omnidirectional desktop CRPZM microphones used in previous experiments. The array was constructed by imbedding the array elements in a 54-cm long and approximately 5-cm high piece of foam rubber which was arranged horizontally and attached to a table stand. The front surface of the elements were flush with the front surface of the foam which provided some acoustic absorption around the array elements.

The CRPZM microphones were arranged such that they were directly underneath the center two array elements. Data were collected for this experiment using 11 input channels of a 16-channel multi-channel A/D converter that provided synchronous sampling of the signals to the various microphones. The 11 channels consisted of the 8-element array, the pair of CRPZM microphones, and the CLSTK microphone for best-case comparisons.

The subjects sat in a chair in the Computer Laboratory at a one-meter distance from the array, wearing the CLSTK microphone. The height of the array was set to be even with the subject's mouth. We collected data from 10 male speakers, each speaking 14 utterances (5 alphanumeric and 9 census utterances). Different speakers were used for training and testing. The training set used was the CRPZM set described in Section 3.5.

7.1.2.2. Experimental Results

Figure 7-2 shows the word recognition accuracy produced by the speech recognizer for this experiment. The word accuracy improves for each successive doubling of sensors and levels off between 4 and 8 elements. Similar results were obtained for the cases of cross-correlation processing

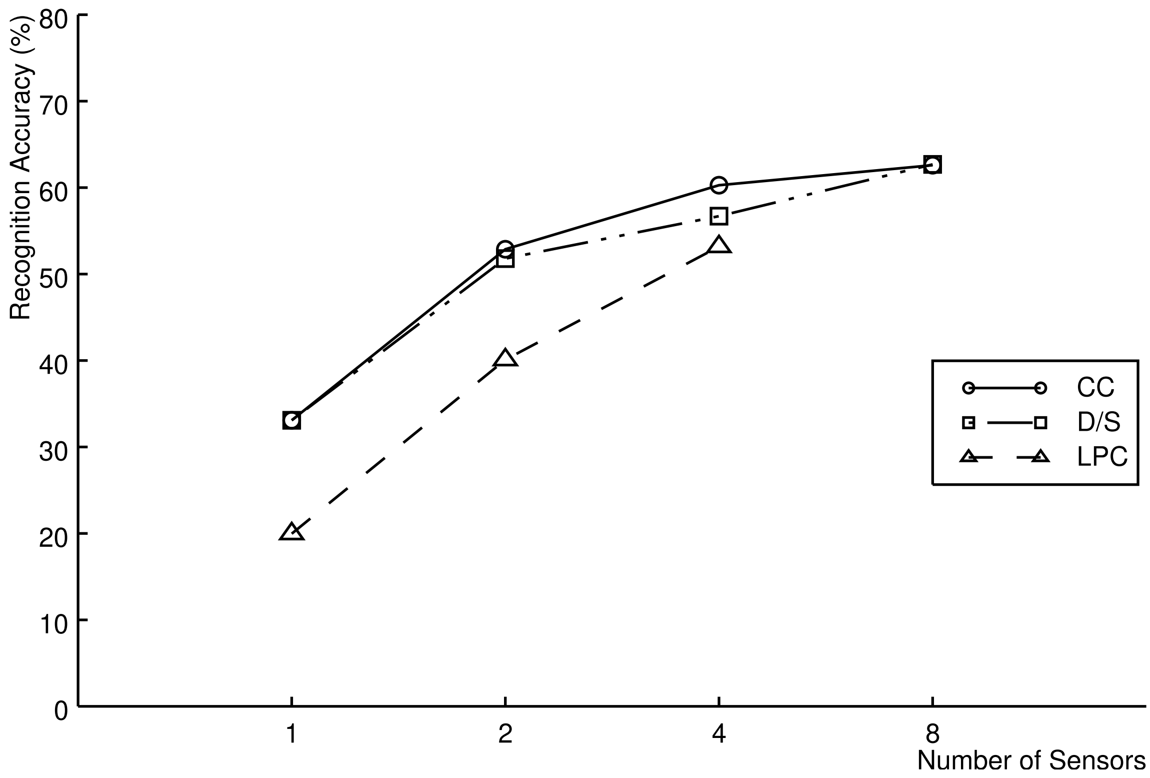

Figure 7-2: Results of experiment using 1-, 2-, 4-, and 8- input sensors.

(CC, circle symbols) and delay-and-sum processing (D/S, square symbols) where the monophonic signal obtained from the delay-and-sum beamformer is presented as a monophonic signal to the cross-correlation algorithm. Results obtained using conventional LPC processing of the monophonic signal obtained from the delay-and-sum beamformer are only shown for up to 4-elements (triangle symbols), but the word recognition accuracy results are worse than the CC and D/S cases.

7.1.3. Implementation of Steering Delays

A set of experiments was conducted to examine the importance of correct input steering delays on recognition accuracy. This examination of the effect of steering delays was motivated by the unexpectedly small increase in gain between the 4- and 8- sensor cases in the experiments on the number of sensors in Section 7.1.2, as well as the relatively small improvement observed for the correlation-based processing compared to simple delay-and-sum processing.

7.1.3.1. Experimental procedure

The training and testing data for these experiments were the same as in the previous experiment. We used the Seneff filterbank and a squaring half-wave rectifier as well.

Our initial experiments assumed that the incoming desired signal is directly on-axis to the array and would arrive at all sensors simultaneously with no delay in arrival time between adjacent sensors. This would be true if the desired source signal was propagated as a plane wave, but this does not occur when the speaker is only 1 meter from the array. In such a situation, sound propagation is more accurately modeled by a point source (the near-field approximation). The general goal of steering delays is to compensate for the differences in arrival time of the desired signal to the various sensors in the delay. Steering delays are inserted at the point of A/D conversion.

7.1.3.2. Types of Steering Delays

Three configurations were used in experiments to test the effects of steering delay on recognition performance:

- **No steering delays:** With this configuration, we assume that the desired signal is propagated from the source as a plane wave, so the desired signal arrives on-axis to the array and at the same time to all sensors.

- **Variable steering delays:** In this case we assume that the desired signal is propagated from the source as a point source such as a spherical wave, but that the location of the source is not known ahead of time. For this case, it is necessary to use a source localization algorithm to find the delays in signal arrival between sensors. The delays are then applied to align all of the sensors such that the desired signal is in phase in all sensor channels (after the delays) prior to passing the signal into the filterbank section of the processing.

- **Hard-wired steering delays:** As with the variable steering delay method, we assume the signal emanates as a point source. We also assume that the source of the signal is located along the perpendicular bisector to the center of the array at a known distance (which is dependent on the particular experiment). With this information, the appropriate compensating delays can be pre-calculated using elementary trigonometry.

For our array of 8 elements spaced 7 cm apart, if the speaker is sitting approximately one meter from the array, using an upsampling factor of 8, the hardwired steering delay times and number of [upsampled] samples would be (using 344 m/s as the speed of sound):

- •Elements 1 and 8 (24.5 cm from the center): 0.086 ms or ~11 samples.
- •Elements 2 and 7 (17.5 cm from the center): 0.044 ms or ~6 samples.
- •Elements 3 and 6 (10.5 cm from the center): 0.016 ms or ~2 samples.
- •Elements 4 and 5 (3.5 cm from the center): 0.002 ms or ~0 samples.

It is obvious from these calculations that if the input signal is indeed better modelled as a point source than a plane wave, then applying steering delays will be necessary prior to processing the sensor input signals even when the subject remains exactly on-axis.

In either of the latter two cases, application of the correct steering delay may require upsampling the input signals, depending on the input sampling rate and the spacing of the array sensors. If the delay between adjacent sensors is some fractional value of the sample period, it is necessary to increase the sampling resolution in order that sub-sample periods are available to be used for delay times. In our experiments we upsampled by a factor of 8 to allow for fractional sample alignment. As the spacing between adjacent sensors gets smaller, or a signal is arriving with a small phase difference between two sensors, higher sampling resolution may be necessary to determine and to implement those time delays.

7.1.3.3. Experimental Results

In order to determine appropriate values for variable steering delays, we first estimated the source location of the speaker by cross-correlating adjacent pairs of input signals and searching for the peak in the cross-correlation function. Because our talkers were seated directly in front of the array (on-axis to the center of the array), the recognition accuracy obtained using variable delays was very similar to the accuracy obtained using pre-calculated hard-wired delays.

Figure 7-3 shows the results of our experiments on steering delays. We found that the difference in performance between not using any steering delays and using either hardwired or variable steering delays was substantial (a gain of approximately 6% in absolute recognition correct accuracy). The 8-element array yielded a 56% word accuracy when using no steering delays, 60% word accuracy using hard-wired steering delays with no upsampling, and approximately 62% word accuracy when using either hard-wired delays or automatic localization with upsampling.

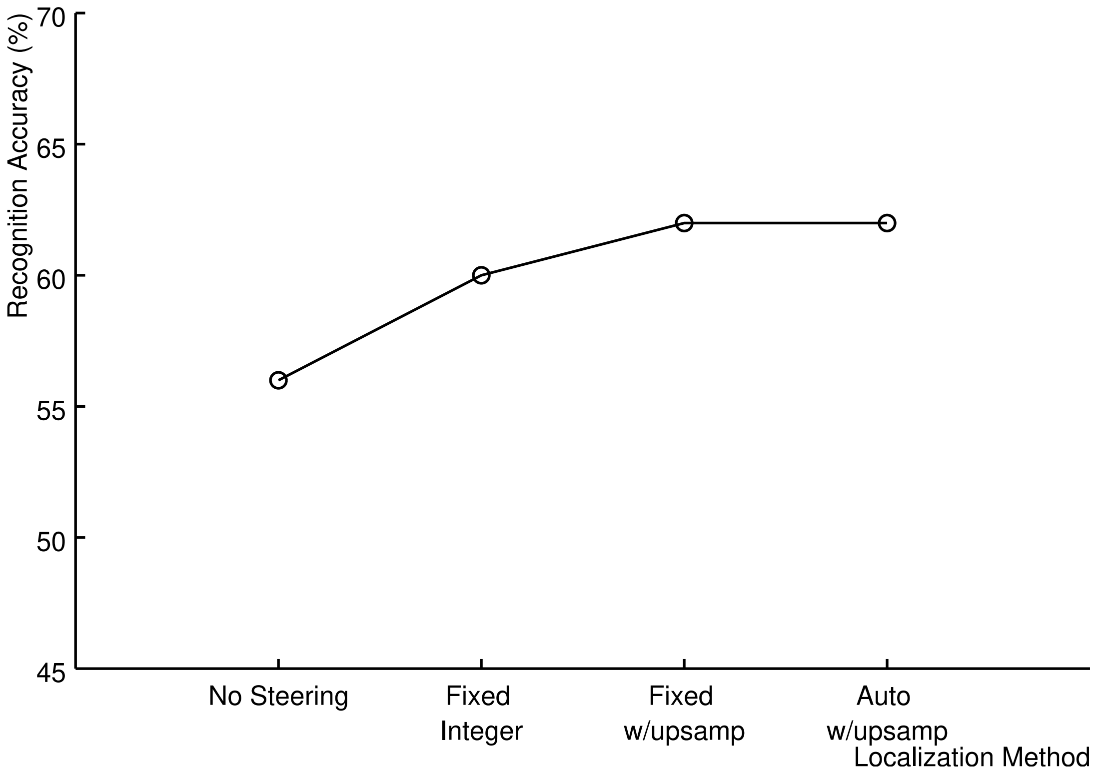

Figure 7-3: Results of experiments using different localization methods for calculating steering delays, and their effect on recognition accuracy.

7.1.4. Effects of Microphone Spacing and the Use of Interleaved Arrays

The array of Flanagan *et al.* [1985] discussed previously in Chapter 4 used different sensor spacings for different frequency bands in order to preserve approximately-constant beamwidth over a range of frequencies. In the experiments described in this chapter we had restricted ourselves to a fixed spacing of 7 cm between elements of our linear arrays. This was done to maintain consistency with the maximum spacing of a pair of CRPZM microphones for comparative testing. As noted previously, with a 7 cm spacing we lose the ability to track the location of signal frequencies above about 2450 Hz if these signals arrive from an angle of 90 degrees off-axis to the perpendicular bisector to center of the array.

In this section we consider the effects of varying the array spacing, and we compare the recognition accuracy obtained using constant array spacing to that obtained using the interleaved array elements introduced by Flanagan *et al.* [1985].

7.1.4.1. Experimental Procedure

As noted above, the array of Flanagan *et al.* [1985], used different sensor spacings for different frequency bands, sharing sensors for the different bands to the extent possible. Since our experimental apparatus only allows us to sample up to 16 microphone channels at once, we chose an interleaved structure of three interleaved arrays each with 7 linearly spaced elements to yield a total of 15 array elements (see Figure 4-1 in Chapter 4). Small spacing is better for high-frequency resolution to reduce spatial aliasing, but small element spacing makes the detection of low-frequency phase differences more difficult. Wider spacing is better for lower-frequency resolution. The sixteenth A/D input channel was used for a CLSTK microphone to provide baseline test results for the collected data.

We constructed two different interleaved arrays, one with linear sub-array spacings of 3, 6, and 12 cm between adjacent elements and one with spacings of 4, 8, and 16 cm between adjacent elements. Minimum frequencies beyond which spatial aliasing occurs (for signals arriving 90 degrees off-axis from the perpendicular bisector to the array) for each spacing is:

- approximately 5733 Hz for 3 cm, 2867 Hz for 6 cm, and 1433 Hz for 12 cm
- approximately 4300 Hz for 4 cm, 2150 Hz for 8 cm, and 1125 Hz for 16 cm

With simultaneous data collected from three linear arrays of varying spacing, we can now compare recognition performance as the spacing between adjacent elements changes. We also can compare the 7-element linear array performance with that of a single array element, and with an entire 15-element interleaved array.

For the 15-element interleaved array experiments, each linear sub-array was used only to calculate the spectral energy values for the frequency bands for which it was best suited. For the 3-, 6-, and 12-cm cases, the 3-cm array was used only for the 2800-8000 Hz (high) range, the 6-cm array for the 1400-2800 Hz (mid) range, and the 12-cm array for the 0-1400 Hz (low) range. Similarly, for the 4-, 8-, and 16-cm array, the 4-cm array was used only for the 2000-8000 Hz (high) range, the 8-cm array for the 1000-2000 Hz (mid) range, and the 16-cm array for the 0-1000 Hz (low) range.

Data were collected using each of the interleaved arrays with the subject sitting at a distance of one meter from the center element of the array. The array height was located at the height of the subject's mouth and was arranged such that the array elements extended horizontally. 14 utterances were collected for each speaker, 5 alphanumeric and 9 census utterances as in the previous experiments. The SNR from a single element was calculated to be approximately 7 dB for the interleaved array with 3-cm minimum spacing (40.7 dB for CLSTK) and approximately 4.2 dB for the interleaved array with 4-cm minimum spacing (40.25 dB for CLSTK).

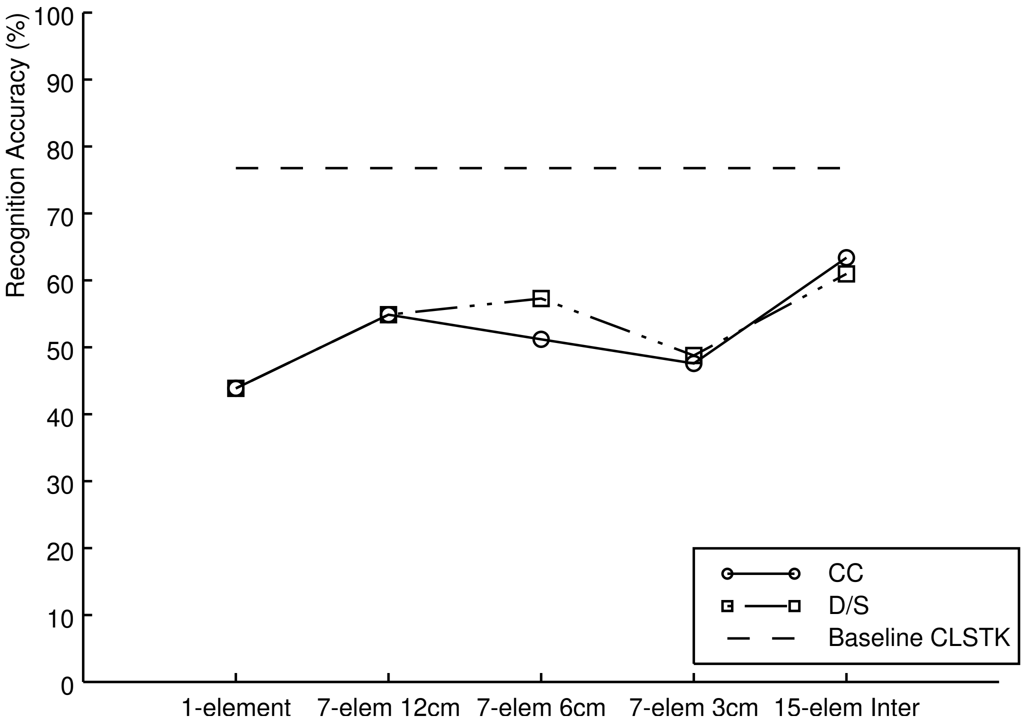

Figure 7-4: Recognition accuracy obtained using arrays with the 3-cm minimum spacing.

7.1.4.2. Experimental Results

Figure 7-4 shows the word recognition accuracy for experiments conducted with data collected using the arrays with 3-cm minimum spacing, and Figure 7-4 shows the corresponding recognition accuracy using the arrays with 4-cm minimum spacing. Both figures show results for cross-correlation processing (CC, circles) and delay-and-sum processing (D/S, squares). The dashed line at the top of the figures shows the recognition accuracy obtained for the simultaneously-recorded speech collected using the CLSTK microphone as a control. For each of the individual 7-element array results, the array was used for the entire frequency range of the input speech. For the case of the 15-element interleaved arrays, each of the three sub-arrays processed only frequency components in the Low, Mid, or High band.

The systems were trained using the CLSTK training data described in Section 3.5.

For the results with 3-cm minimum spacing in Figure 7-4, we see that the use of the 15-element interleaved array provides greater recognition accuracy compared to that obtained with any of the individual 7-element arrays. This is true for both the cross-correlation and delay-and-sum processing. We also see that the results for the individual 7-element array spacings do vary. For the cross-correlation processing, the array spacing used for the mid-frequencies (6 cm) provides the best recognition accuracy. For the delay-and-sum processing, the spacing for the low frequencies (12 cm) is better. The poorest array recognition comes from the most narrow spacing in both cases. This may be due to an inability of our processing to align well to the arrival time differences between input sensors of lower frequencies with such a small sensor spacing.

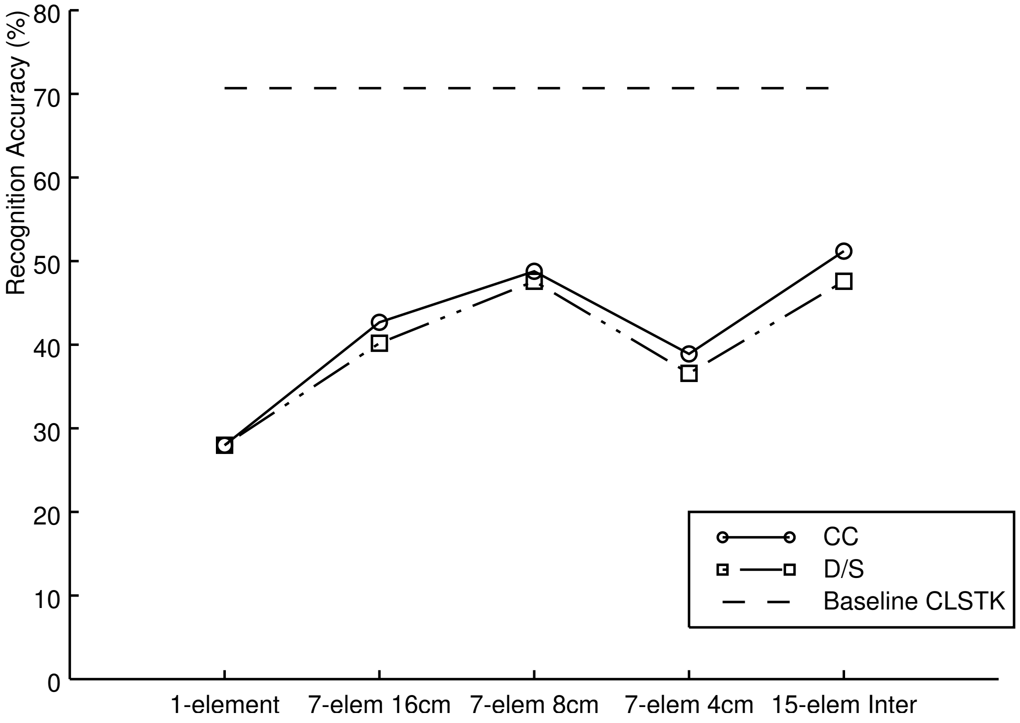

Figure 7-5: Recognition accuracy obtained using arrays with the 4-cm minimum spacing.

For the results with the 4-cm minimum spacing in Figure 7-5, we also see that the use of the 15-element interleaved array provides greater recognition accuracy compared to any of the individual 7-element arrays. Once again this is the case for both the cross-correlation and delay-and-sum processing. Unlike the 3-cm minimum spacing array configuration results, the best recognition is provided in both cross-correlation and delay-and-sum processing by the array spacing used for the mid-frequencies (8 cm). Once again, the 7-element array with the minimum array spacing provides the worst recognition accuracy if used over the entire frequency range.

The data for this experiment suggests that using an interleaved array does provide improved performance over selecting a constant array spacing for the entire frequency range of interest. However, if the increased processing required for interleaving is not an option (due to the multiple array spacings and increased number of input microphones) and only one spacing is to be chosen, the 6-8 cm range was found to provide the best overall performance of the individual spacings.

7.1.5. Effect of the Shape of the Peripheral Filterbank

The separation of the input speech signals into a set of bandlimited signals to isolate individual frequency components prior to cross-correlation is one of the most important aspects of the cross-correlation algorithm. In this section we compare the recognition accuracy obtained using two different filterbanks.

7.1.5.1. Experimental Procedure

We used the half-wave square-law rectifier, along with the data collected using the 15-element interleaved array with the 4-cm minimum spacing. The steering delays were variable, and they were calculated automatically using upsampling by a factor of 8.

The first filter bank tested was the Seneff filterbank as described in Chapter 5 (Section 5.2.3). The Seneff filterbank has 40 frequency bands with spacing based on the filter spacing in the human auditory system. The other filterbanks tested were two types of 40-channel Mel frequency filterbanks (also described in Section 5.2.3). As noted in Chapter 5, the first of these filters has 13 linearly-spaced channels, each with 133.33-Hz bandwidth, overlapped by 66.67 Hz, and 27 log-spaced channels (with a center frequency ratio of 1.071). The second Mel filterbank has 15 linearly-spaced channels, each with 100-Hz bandwidth, overlapped by 50 Hz, and 25 log-spaced channels (with a center frequency ratio of 1.085).

7.1.5.2. Experimental Results

Figure 7-6 shows the results of speech recognition experiments using each of the three filterbanks. Recognition accuracy appears to be very similar for all three of the filterbanks. The recognition accuracy observed for a single array element is much less than that of the baseline CLSTK microphone in part because the SNR of the signal is lower, and in part because there is a greater mismatch between training and testing environments. The fact that similar recognition rates are obtained using the single array element accuracies for all of the filterbanks suggests that the filterbank type may be unimportant for monophonic processing. The Seneff filterbank does provide a substantial improvement in recognition accuracy over the Mel-frequency filterbanks for the 15-element interleaved array. It is unknown whether this is due to the filter shape or filter spacing of the Seneff filterbank compared to the Mel-frequency filters. A set of DFT frequency bin weighting functions (*a la* the Mel-frequency filterbanks) could be designed which approximate the Seneff filterbank to test the shape vs. spacing.

Because of the superior performance of the Seneff filterbank in these comparisons, it will be used as the default filterbank for our remaining experiments.

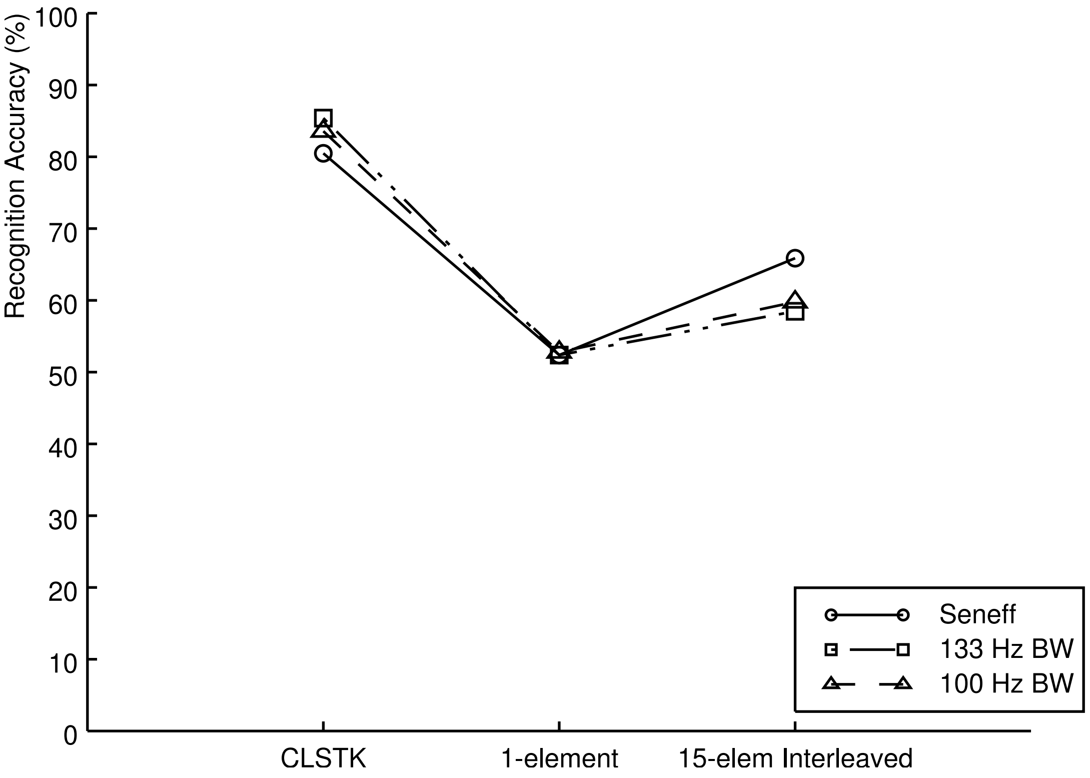

Figure 7-6: Word recognition accuracy for three tested filterbanks.

7.1.6. Reexamination of Rectifier Shape

Because of the many changes in system configuration since the original rectifier experiments in Section 7.1.1, we repeated the experiments examining the dependence of recognition accuracy on shape, but with the current “best” configuration of the cross-correlation-based system.

7.1.6.1. Experimental Procedure

The full 15-element interleaved array with the 4-cm minimum spacing was used in this experiment. The Seneff filterbank was used for the filtering and automatic localization was enabled. The training data are the CLSTK data described in Section 3.5.

We compared recognition accuracy obtained using the Seneff rectifier and the halfwave power-law rectifiers using exponents of 1 through 5.

7.1.6.2. Experimental Results

Figure 7-7 shows word recognition accuracy obtained for four sets of speech: a) speech recorded monophonically using the CLSTK microphone (circles), b) a single array element (squares), c) the best 7-element linear sub-array (triangles) which in this case had the 8-cm element spacing, and d) the 15-element interleaved array (bullets).

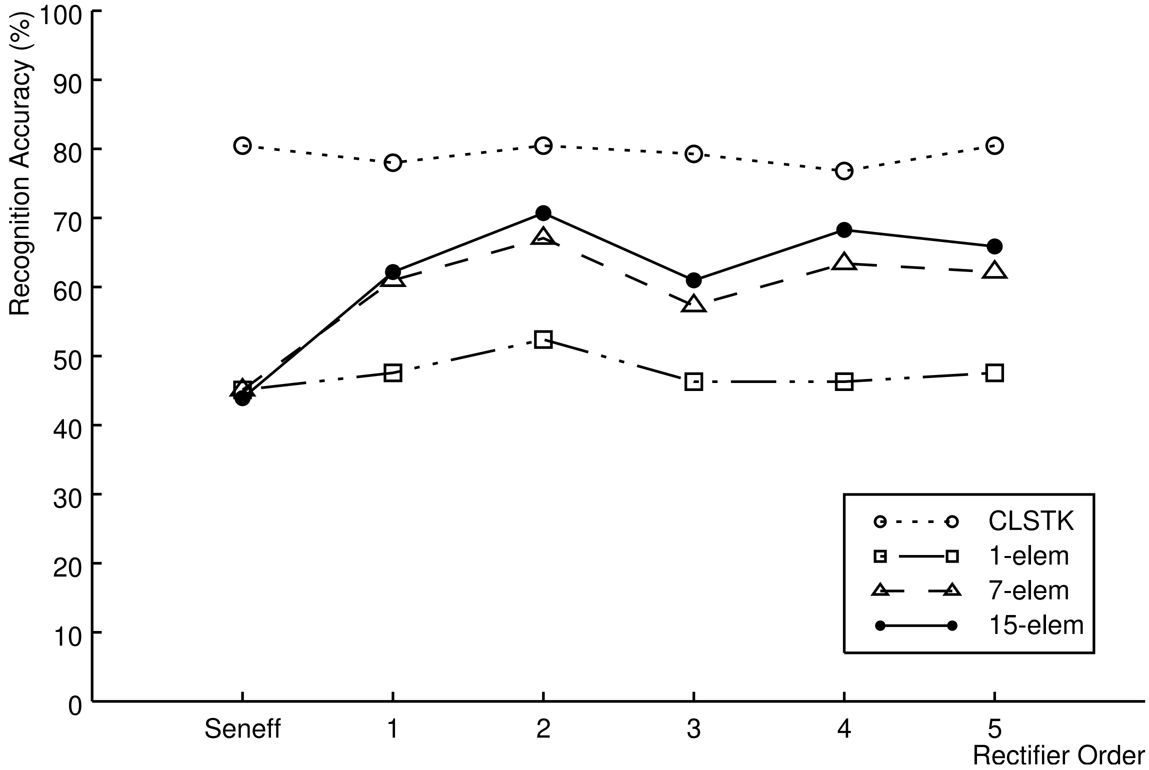

Figure 7-7: Effect of rectifier type on recognition accuracy using larger arrays.

Our findings basically re-confirmed our earlier results from Section 7.1.1. In all four testing configurations we found:

- The halfwave square-law rectifier provides the best recognition accuracy of all of the rectifiers tested.

- The Seneff rectifier provided the worst recognition accuracy of all of the rectifiers tested (excluding speech from the CLSTLK microphone).

7.1.7. Summary

In this section, we have examined the processing sections of the correlation-based processing algorithm. The major findings of these experiments are as follows:

- Arrival time of the desired signal to the sensors cannot be ignored. Steering delays provide a substantial increase in recognition accuracy. We determine steering delays automatically on a per-utterance basis by upsampling the input signals by a factor of 8, cross-correlating between adjacent input microphones, finding the location of the peak in the cross-correlation functions, and applying these delays to the array.

- The type of filterbank (at least considering the ones we tested) did not appear to be too important, but our experiments show the Seneff filterbank to yield slightly better performance than the Mel filterbanks, and we therefore select it to be used in the remainder of our experiments.

- Rectifier shape is important. We found that the halfwave square-law rectifier gives the best performance, and therefore we chose to use it in the remainder of our experiments.

- Array microphone spacing was shown to impact on recognition accuracy. The use of an interleaved array which consists of three linear sub-arrays of different spacings was found to give better recognition accuracy over using any of the spacings individually. The sub-arrays are each responsible for a portion of the overall frequency band of interest, with the smallest spacing for the highest frequency band, the widest spacing for the lowest frequency band, and an in-between spacing for the middle frequency band.

7.2. Comparison of Recognition Accuracy Obtained with Cross-Correlation Processing, Delay-and-Sum Beamforming, and Traditional Adaptive Filtering

7.2.1. Introduction

The recognition accuracy observed in the previous sections using cross-correlation processing was almost always better than all other types of processing considered. Nevertheless, the advantage of cross-correlation processing over delay-and-sum beamforming was frequently rather small. This was disappointing, especially in light of the promising results in the pilot experiments described in Chapter 6.

In this section we describe a series of experiments that examine some of the reasons why the difference in recognition accuracy observed in real speech recognition experiments is less than what we had expected from the pilot experiments. We consider effects of SNR, the use of environmental compensation procedures, and differences between results obtained using speech that is subjected to simulated degradation and speech in real noisy environments.

7.2.2. Initial Results using the Conference Room Environment

The cross-correlation processing "outperforms" delay-and-sum beamforming in the pilot experiments using pairs of tones for the desired signals, as depicted, for example, in Figures 6-2 through 6-5. However, all algorithms will perform poorly if the SNR is sufficiently low. The Conference Room is the second recording environment described at the beginning of this Chapter. Recordings were made in the Conference Room to provide some speech data that were less severely degraded than the speech in the Computer Laboratory.

7.2.2.1. Experimental Procedure

Recordings were made in the conference room environment, as described at the beginning of Chapter 7 above. Alphanumeric and census data were collected with the speaker sitting at a distance of 1 meter and 3 meters from the interleaved array with a minimum spacing of 4 cm. The array was set to be at the height of the speaker's mouth, and the speaker sat on-axis to the center element of the array. The speaker also wore a CLSTK microphone for baseline control data collection.

A set of additional speech samples was collected at the same two distances with an additional jamming signal source. The jamming signal was approximately 45 degrees off axis to the array, at about the same distance (1 or 3 meters, depending on the experiment) from the center element of the array. The source was a monophonic AM radio signal, with the radio tuner set to a talk radio station.

Average SNRs for a single array element from the collected data sets were 18.3 dB for 1-meter with no additional jamming signal, 11.5 dB for 1-meter with the additional jamming signal, 13.9 dB for 3-meter with no additional jamming signal, and 6.8 dB with the additional jamming signal. The SNR for speech recorded using the CLSTLK microphone was approximately 50 dB.

The correlation-based processor used the Seneff filterbank with the halfwave square-law rectifier and automatic localization to calculate the steering delays for the incoming speech signals.

7.2.2.2. Experimental Results

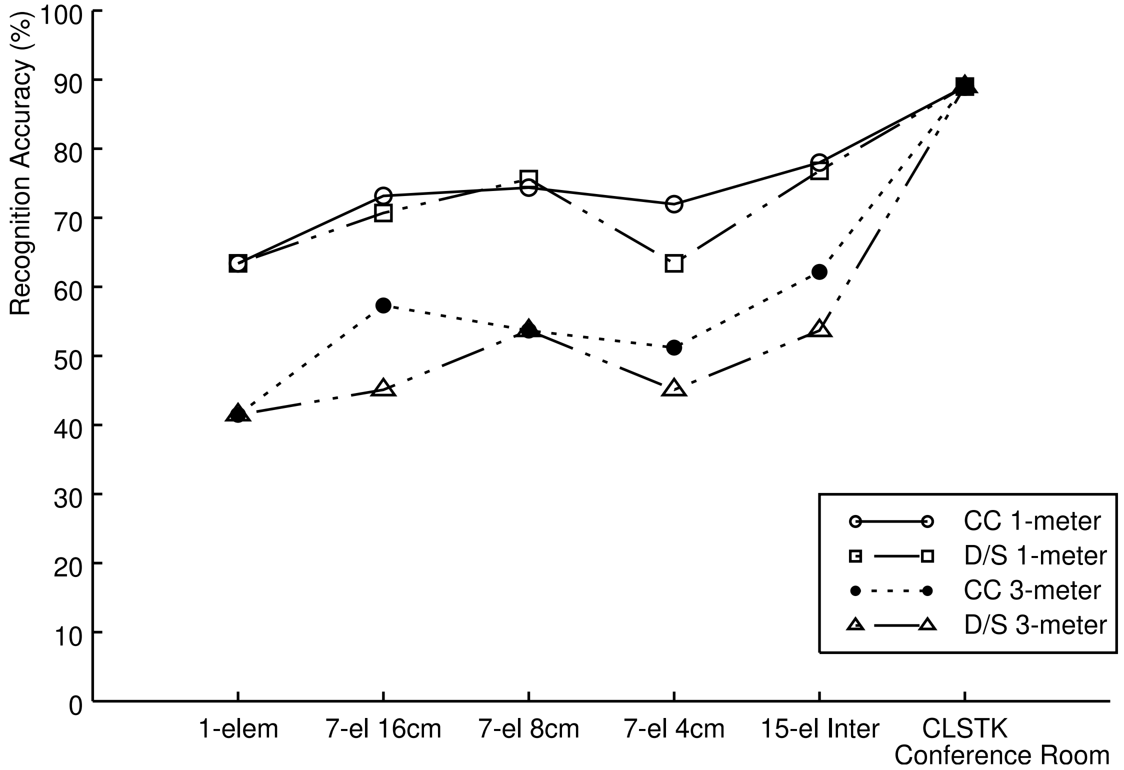

Figure 7-8: Recognition accuracy observed in the conference room at 1 and 3 meters with no jamming

Figure 7-8 compares the recognition accuracy obtained for the 1- and 3-meter data sets collected in the conference room with no additional jamming noise, using various array configurations and the CLSTLK microphone.

The data trends for recordings at the 1-meter distance are very similar to results that were previously observed from data collected in the more noisy computer laboratory (Figure 7-5). With both cross-correlation processing (circles) and delay-and-sum beamforming (squares) the 15-element interleaved array performance is superior to any of the individual 7-element sub-arrays. We also see that the 8-cm array spacing associated with the 7-el Mid case provides the best recognition accuracy of the three 7-element sub-arrays.

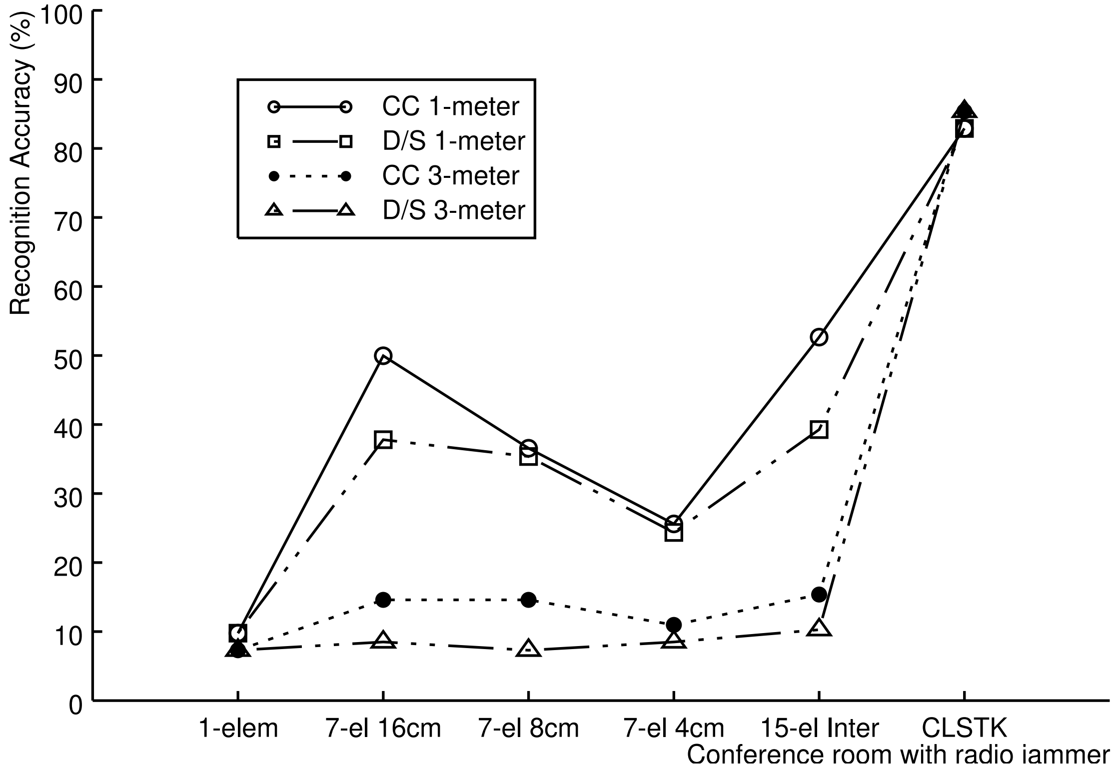

Figure 7-9: Recognition accuracy observed in the conference room with jamming noise added.

When the recording distance is increased to 3 meters, the SNR decreases to the decay of the direct signal power over distance, and due to the increased contributions of the noise from other sources in the room and reflections due to reverberation. Overall recognition accuracy at this distance decreases, but the relative error rates across recording and processing conditions are similar to what had been seen as was seen at the 1-meter distance case. The cross-correlation processing appears to provide a slightly greater advantage in recognition accuracy compared to the delay-and-sum beamformer when the speech is recorded at the 3-meter distance.

Figure 7-9 shows similar comparisons of recognition accuracy as were seen in the previous figure, but for the data that were obtained with the additional jamming talk-radio source.

Again, the cross-correlation processing provides equal or better recognition accuracy than the delay-and-sum beamformer, although neither performs very well at the 3-meter distance. For both distances, best performance is obtained with sub-arrays with wider element spacing. This may be because of decreased SNR in the higher frequencies in the jamming radio signal, even though the actual radio program material is bandlimited to below 5 kHz (the amplifier output of the radio goes beyond that, outputting only noise for frequencies above 5 kHz).

7.2.3. Use of Environmental Compensation Algorithms

Microphone arrays impose spectral coloration on the signals that they process, and the frequency response of an array will depend at least in part on direction of arrival of the desired speech signal. A number of environmental compensation algorithms have been developed to ameliorate the effects of unknown additive noise and unknown linear filtering on speech signals ([Stern *et. al.*,1995], [Juang, 1991]). In this section we describe some results obtained applying the Codeword Dependent Cepstral Normalization (CDCN) algorithm ([Acero,1990], [Moreno *et. al.*, 1995] to the output of our "best" array configuration. The CDCN algorithm attempts to estimate the parameters characterizing the unknown noise and filtering that is presumed to have corrupted a speech signal, and then apply the appropriate inverse signal processing operations. The CDCN algorithm was chosen for this purpose because it provides robustness in a variety of operating environments, and because it does not require that environment-specific "training" data be available *a priori*. These experiments are similar to those we performed at the Rutgers CAIP Center, as described in Chapter 4.

The implementation of CDCN used in these studies was an updated version by Moreno *et al.* [1995]. The algorithm is basically the same as before, except that the testing environment is now mapped directly to a model of the training environment instead having the training and testing environments mapped to some "neutral" model.

7.2.3.1. Experimental Procedure

Cepstra obtained from training data processed using the cross-correlation algorithm are used by the CDCN algorithm to obtain a model of the training environment. Incoming cepstra from testing data for a particular experiment are normalized to most closely resemble cepstra from the training data. We used the CLSTK training data described in Section 3.5 as the training set for these experiments. The test data were recorded in the conference room data using the 15-element interleaved array data recorded with 4-cm minimum spacing. The speakers sat 1 meter or 3 meters from the array.

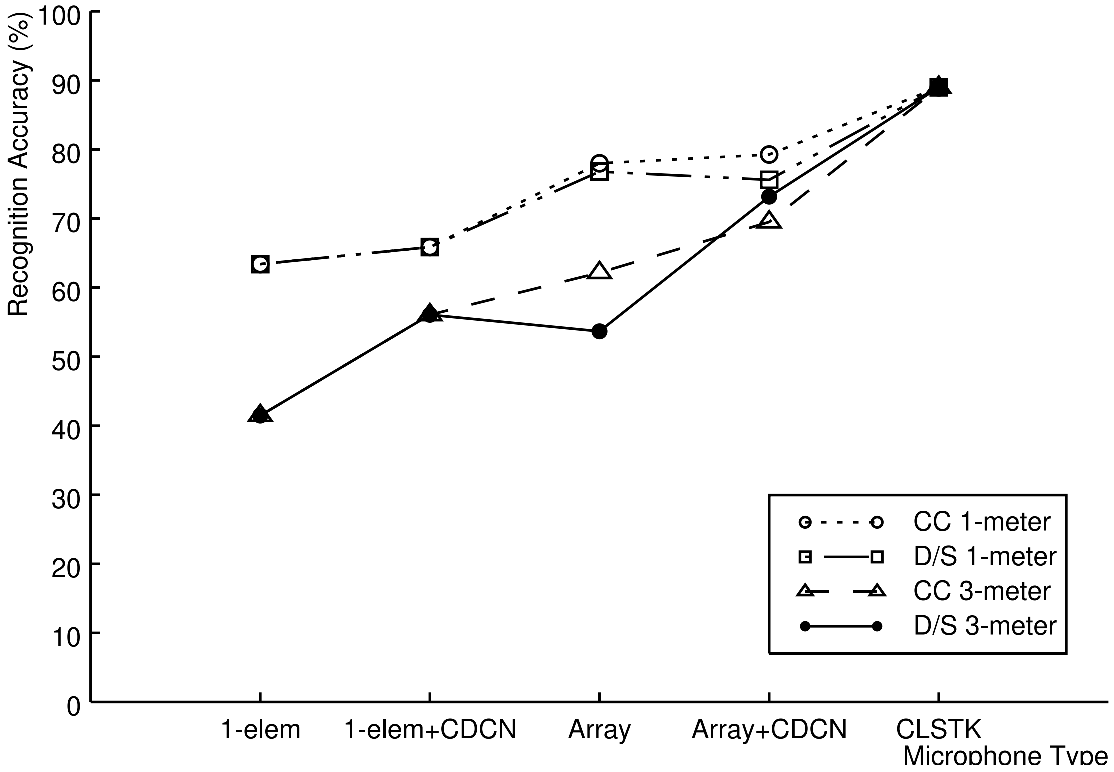

Figure 7-10: Recognition accuracy obtained with and without CDCN compensation.

7.2.3.2. Experimental Results

Recognition accuracy results with and without CDCN are shown in Figure 7-10. As in the case of the data recorded at the Rutgers CAIP Center using the Flanagan array (Chapter 4), we found that use of CDCN increased the performance of the individual array element cases and the full array cases. We also find that the array performance itself is better (in all cases but the delay-and-sum 3-meter case) than results obtained using a single array element with the CDCN algorithm. These comparisons confirm that the use of arrays and environmental normalization algorithms is complementary, both for the correlation-based processing as well as for delay-and-sum array beamforming. Unlike the case of the Rutgers data, however, the combination of the array and CDCN compensation did not increase the recognition accuracy to the level obtained using the CLSTLK microphone. This may be because the Flanagan array used in the Rutgers data had 23 elements (compared to our 15), or because the recording conditions of the conference room may be more difficult than those at Rutgers.

7.2.4. Comparison with Other Array Algorithms Using Artificially-Added Noise

In the introductory sections we noted that there are three approaches to microphone array processing: delay-and-sum beamforming, traditional adaptive filtering, and the cross-correlation-based processing. In our experimental comparisons thus far we have focussed on delay-and-sum beamforming and the cross-correlation-based processing, with little attention paid to traditional adaptive filtering methods. We did not expect traditional adaptive filtering approaches to perform well in environments in which reverberation as well as additive noise is a factor, as such environments violate the assumption that the desired signals and sources of degradation are statistically independent.

In this section we describe the results of several experiments comparing the recognition accuracy obtained using all three array-processing methods when the signal is “clean” speech artificially degraded by additive noise. The artificially-added noise was used because it provides the means to obtain a more controlled set of stimuli than would be possible using natural speech data. In the experiments in this and the following section we are concerned with comparisons of recognition accuracy obtained using simulated versus real degradations, as well as comparisons of correlation-based processing with delay-and-sum beamforming and traditional adaptive filtering.

We used the Griffiths-and-Jim algorithm [Griffiths and Jim, 1982] as the archetypal adaptive filtering algorithm, in part due to its relative ease of implementation within our framework, and because it has the interesting property that it reduces to a delay-and-sum beamformer if the weights of the adaptive filters are all constrained to be zero. Updating the filter taps of the Griffiths-and-Jim processor is only performed when the desired speech signal is not present. *i.e.* only frames of the input utterance containing noise alone.

7.2.4.1. Experimental Procedure

To verify our Griffiths-and-Jim implementation and to provide a baseline for further experimentation, we conducted an experiment in which we corrupted the CLSTK data from one of our previous experiments with additive white Gaussian noise. Figure 7-11 shows the recognition accuracy for monophonic CLSTK files (40.25 dB SNR) corrupted with additive white noise in a range of SNRs from 6 to 31 dB.

For Figures 7-12 and 7-13, array-like test signals were created by delaying the noise in successive amounts to each of six "sensors" prior to adding it to the CLSTK "on-axis" signal, making a 7-element multi-channel input signal file by interleaving these signals. This noise was added to the CLSTK signal to provide SNRs of 6 dB and 11 dB. Correlation-based processing, delay-and-sum processing, and Griffiths-and-Jim processing were then applied to the 11-dB SNR and 6-dB SNR cases prior to running the signals through the SPHINX system.

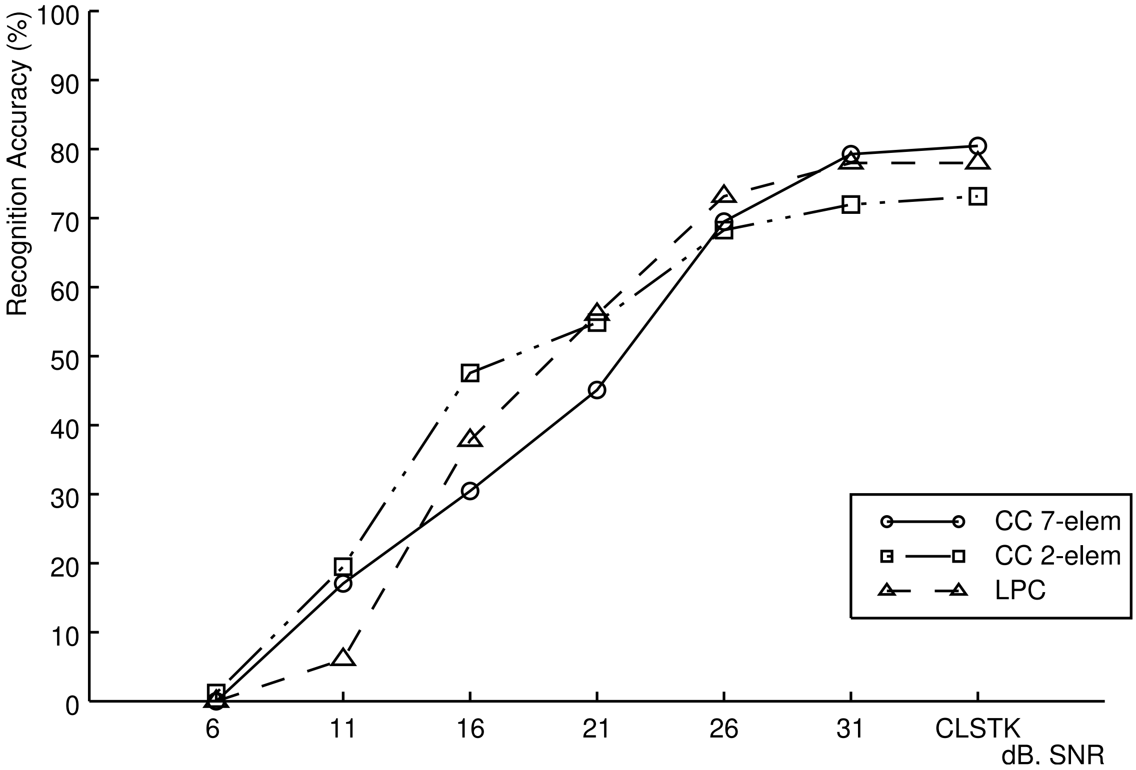

Figure 7-11: Recognition accuracy obtained by adding noise to "monophonic" signals at various SNRs.

7.2.4.2. Experimental Results

Figures 7-11 through 7-13 show the results of these experiments. In Figure 7-11, recognition accuracy for "monaural" signals (with both speech and noise signals arriving at all array elements simultaneously) is plotted as a function of SNR. Figures 7-12 and 7-13 show recognition accuracy at two fixed SNRs, 11 and 6 dB, respectively, as a function of the amount by which the added noise is delayed from element to element of the array.

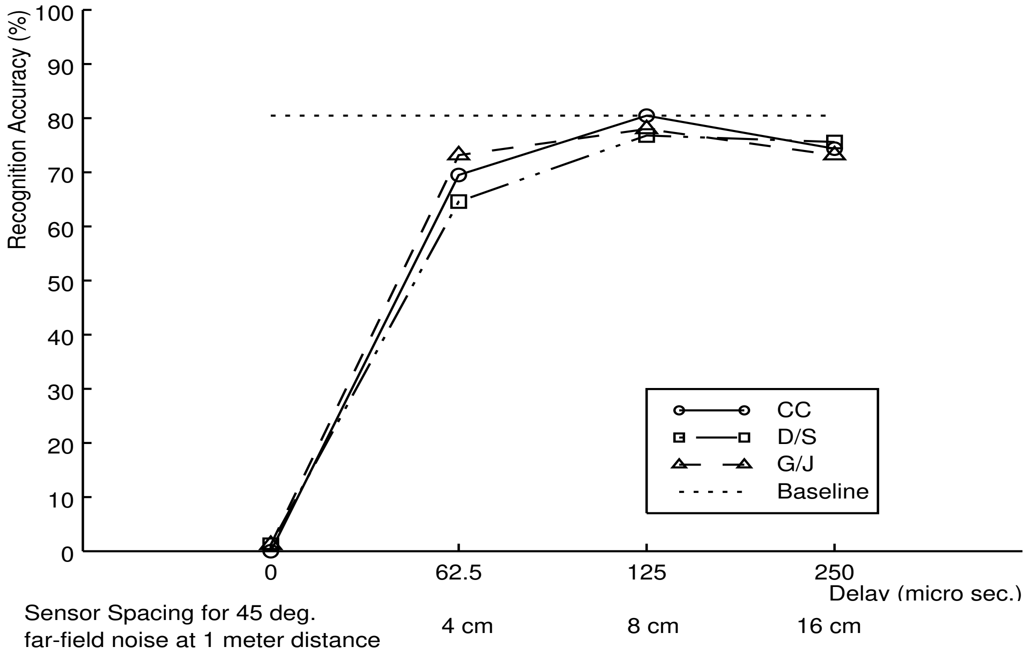

Figure 7-12: Word recognition accuracy for array processing algorithms with noise added at an SNR of 11 dB. Results are plotted as a function of delay of the added noise from element to element of the array.

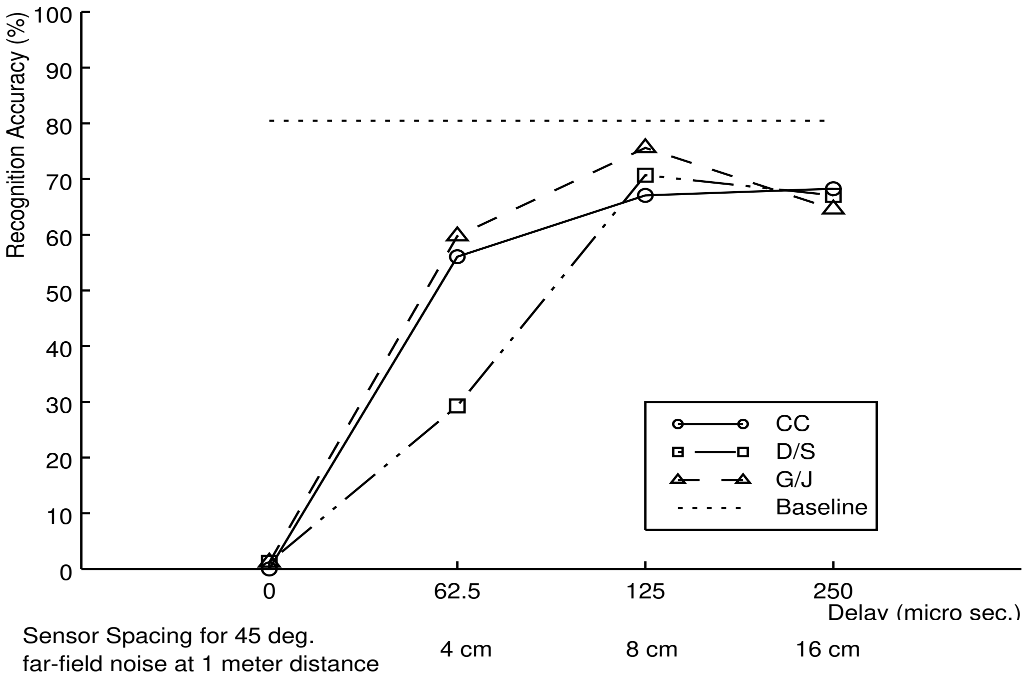

Figure 7-13: Word recognition accuracy for array processing algorithms with noise added at an SNR of 6 dB. Results are plotted as a function of delay of the added noise from element to element of the array.

Unsurprisingly, we observe that the recognition accuracy for the single-element cases decreases as SNR decreases (Figure 7-11). In cases where the noise signals are delayed from element to element of the array, the correlation-based and delay-and-sum processors provide almost equal amount recognition accuracy while the Griffiths-and-Jim processing provides slightly better recognition accuracy. This confirms that the Griffith-and-Jim processing can be helpful in purely additive noise environments. The next experiment compares the recognition accuracy when applying these three array processing algorithms to real array data.

7.2.5. Comparison with Other Array Algorithms Using Real Environments

We now apply the Griffith-and-Jim algorithm to the array data collected in both the computer laboratory and the conference room environment. Figure 7-14 shows the relative performance of the Griffiths-and-Jim processing vs. correlation-based and delay-and-sum processing for speech collected in the computer laboratory using the 15-element interleaved array with both the 3-cm and 4-cm minimum spacing. Figure 7-15 compares the recognition accuracy obtained with the three processing algorithms using the 4-cm minimum spacing conference room data at the 1- and 3-meter distances with no additional jamming, while Figure 7-16 compares the recognition accuracy obtained with the three processing algorithms using the 4-cm minimum spacing conference room data at the 1- and 3 meter distances with the additional jamming by the radio talk show.

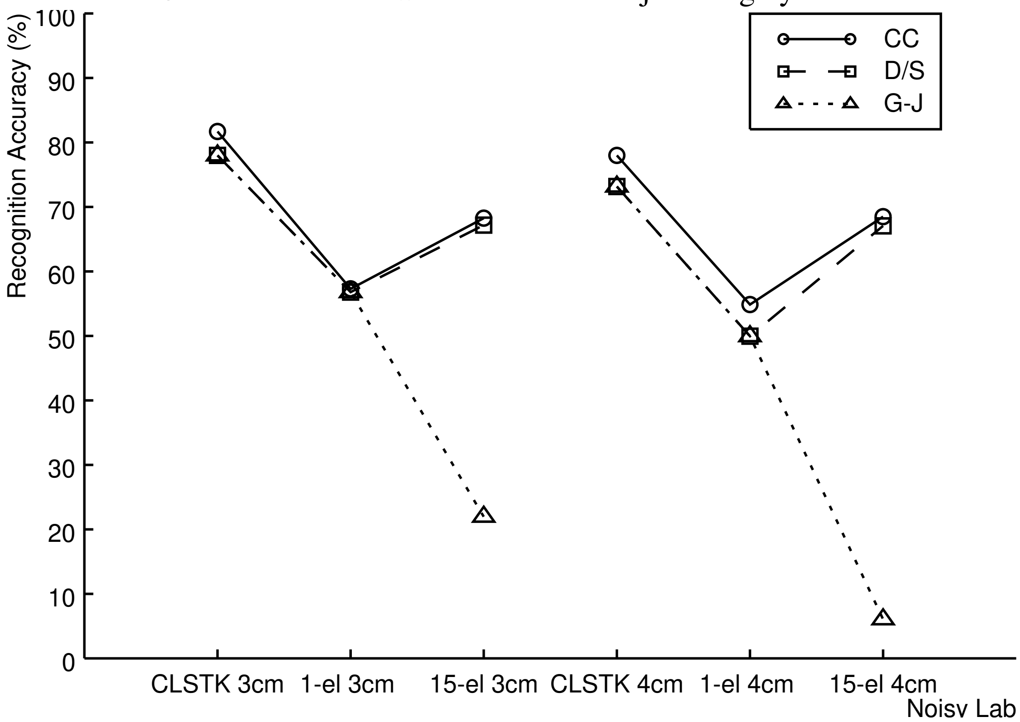

Figure 7-14: Recognition accuracy obtained using three array processing algorithms for data recorded in the computer laboratory.

With the relatively low SNR of the computer laboratory, we find (Figure 7-14) that the Griffiths-Jim algorithm (triangles) fails miserably compared to the relatively equivalent recognition accuracy provided by the cross-correlation based processing (circles) and the delay-and-sum processing (squares). In both cases the 15-element interleaved array exhibits better recognition accuracy compared to a single array element. The low SNR makes it difficult to detect the difference between speech and non-speech portions of the input signals, which is necessary for our Griffiths-Jim implementation to avoid signal cancellation. It may be the case that the poor performance of the Griffiths-Jim algorithm in the noisy computer lab environment is due to signal cancellation problems.

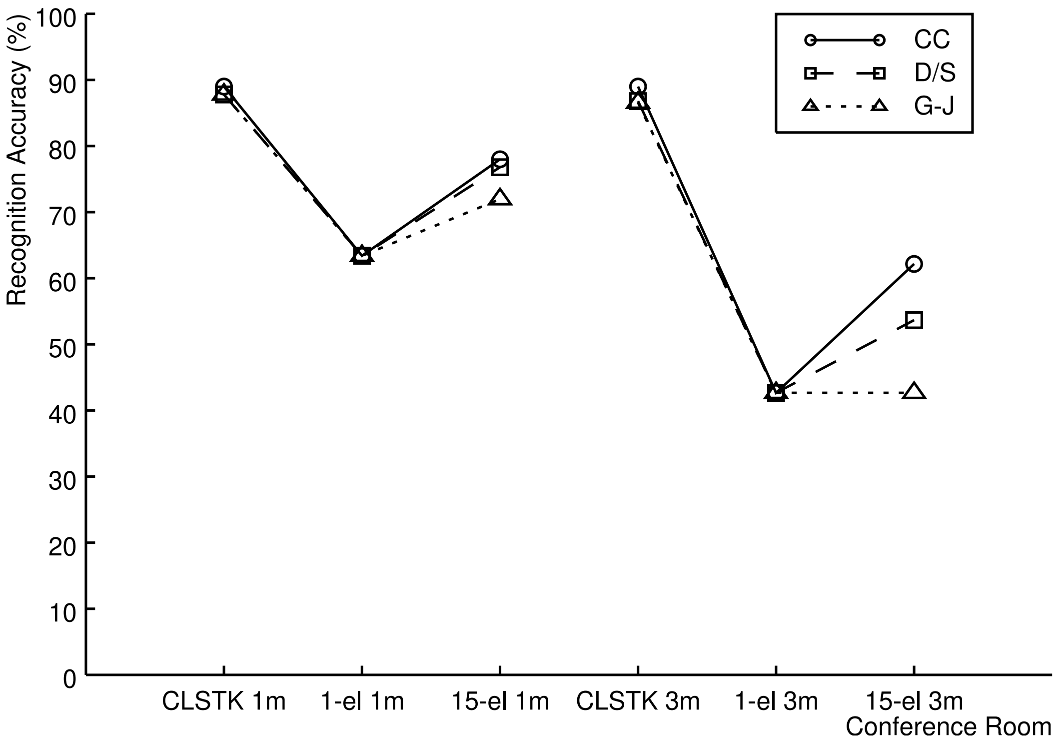

Figure 7-15: Recognition accuracy obtained using three array processing algorithms for data recorded in the in the conference room with no jamming signal.

The recognition accuracy results in Figure 7-15 and Figure 7-16 show that the Griffiths-and-Jim processing (triangles) does not fare as well as either the correlation-based processing (circles) nor the delay-and-sum processing (squares) for data collected in environments where the corrupting sources aren't merely additive. The recognition accuracy at 1 meter in the conference room with no jamming noise is not much worse than what is observed with the delay-and-sum beamforming, but performance decreases more dramatically as the distance from the desired source to the array increases with no jammer or if a jamming noise is introduced (Figure 7-16).

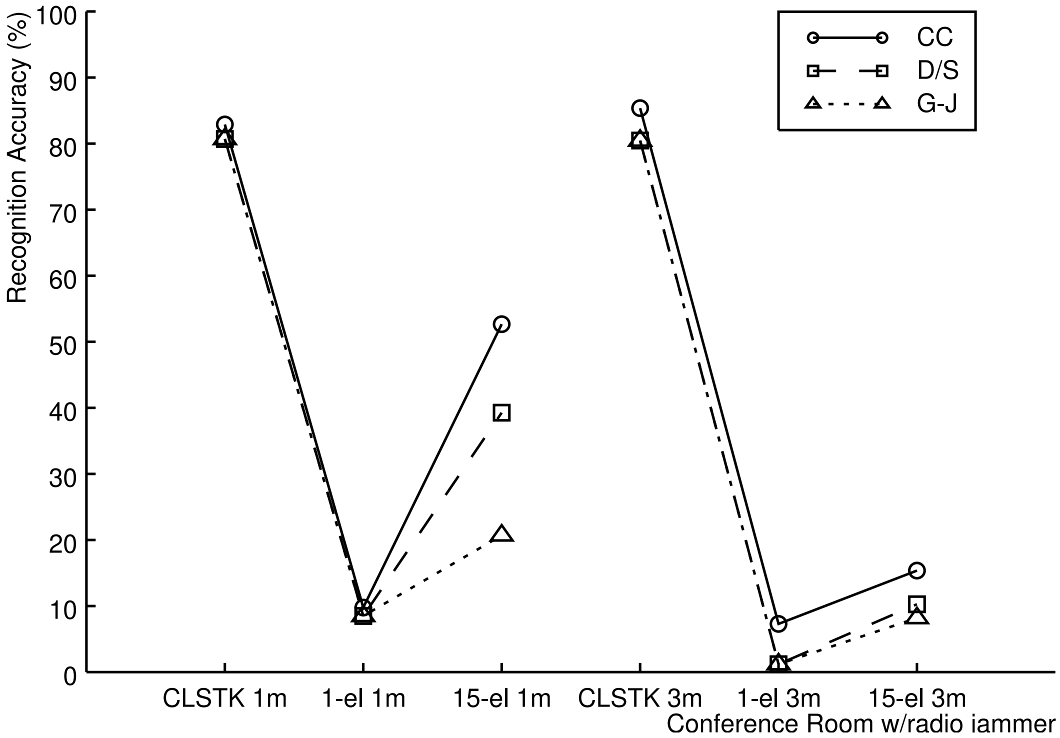

Figure 7-16: Recognition accuracy obtained using three array processing algorithms for data recorded in the in the conference room with a talk radio jamming signal.

7.2.6. Comparison of Results Using Artificially Added Noise to Results Obtained in Real Environments

In this section, we provide a comparison between the results of experiments conducted by adding noise artificially to clean speech signals vs. results obtained in real environments. This is an important comparison because it shows that what often provides improvement to artificially corrupted signals does not necessarily translate to similar results on signals in real world environments.

Figures 7-17 and 7-18 compare results using signals with artificially-added noise to results obtained in the conference room (a “real” environment). In Figure 7-17, the comparison is between the additive noise added at 11 dB (from Figure 7-12) vs. the 4-cm minimum spacing array data collected in the conference room environment with the radio jamming signal at a distance of 1-meter

(11.5 dB SNR). In Figure 7-18, the comparison is between the additive noise added at 6 dB (from Figure 7-13) vs. the 4-cm minimum spacing array data collected in the conference room environment with the radio jamming signal at a distance of 3-meters (6.8 dB SNR). These data sets were chosen for these comparisons because of the relatively close SNRs of the sets.

The figures show the results for each data set processed with the three processing methods compared in this chapter: the Griffiths-Jim beamformer, the delay-and-sum beamformer, and the cross-correlation processing algorithm. For all three processing methods, we observe that clean speech corrupted by artificially added noise provides a much higher recognition accuracy than is obtained with speech collected in a real environment at a similar SNR. The conclusion here is that we must be careful when interpreting data obtained by corrupting speech with artificially added noise.

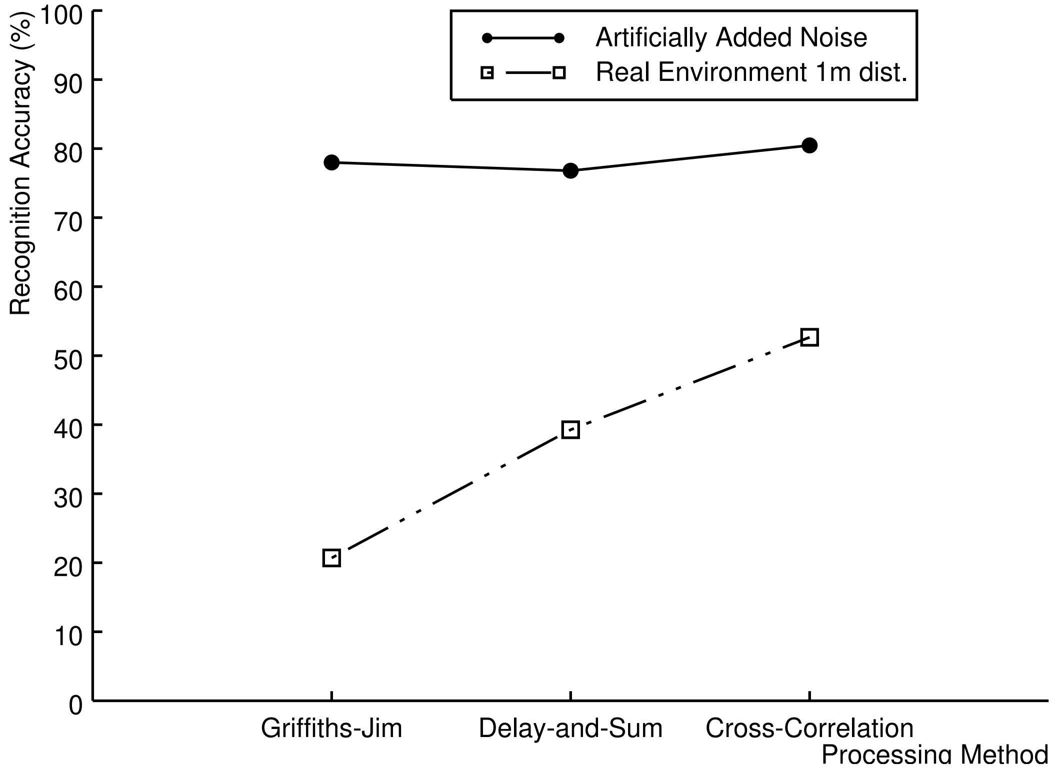

Figure 7-17: Comparison of recognition accuracy obtained by processing clean speech corrupted with artificially added noise at 11 dB SNR vs. recognition accuracy obtained by processing speech with a SNR of 11.5 dB collected in the conference room with a talk radio jamming signal present.

It appears that differences between testing and training microphone conditions, reverberant rooms, and correlated noises pose greater difficulty for the Griffiths-and-Jim algorithm than for either the correlation-based processing or the delay-and-sum processing. These results concur with past researchers' findings (Chapter 2) in which traditional LMS beamforming algorithms were used in reverberant environments (*e.g.* [Peterson, 1989], [Van Compernolle, 1990]). Traditional adaptive filtering algorithms have not proven to be much help when noise sources present at the array inputs are correlated with the desired signal. We don't understand why the results using the Griffiths-Jim algorithm in the noisy lab are so much worse than the results from the conference room environment, as the conference room appears to be more reverberant. We hypothesize that the severe degradation is due to signal cancellation problems resulting from an inability to faithfully detect in an automatic manner speech and non-speech portions of the input signal in such a low SNR environment.

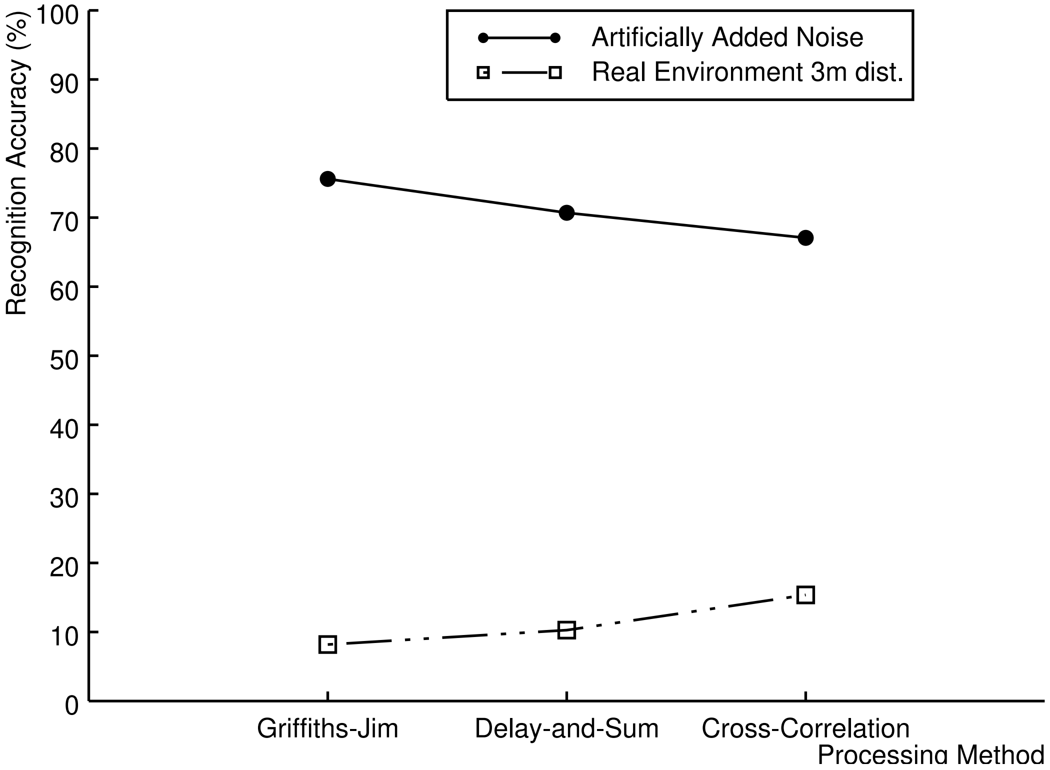

Figure 7-18: Comparison of recognition accuracy obtained by processing clean speech corrupted with artificially added noise at 6 dB SNR vs. recognition accuracy obtained by processing speech with a SNR of 6.8dB collected in the conference room with a talk radio jamming signal present.

7.3. Computational Complexity

In this section we address the subject of computational complexity. One of the biggest application-oriented drawbacks to our correlation-based system is that it is extremely computationally expensive, limiting its usability to non-real-time systems. Table 7-1 shows the computational requirements of these forms of array processing for one millisecond of input speech. In the table, the amount of computation required to implement the correlation-based processing(CC), delay-and-sum beamforming(D/S), and Griffiths-Jim beamforming (G-J) algorithms are compared. The amount of processing necessary for a monophonic input signal to the SPHINX-I LPC-based processing is included as well.

In a system utilizing the correlation-based processing introduced in this thesis, for each of the K input microphone channels (calculations are for K=7 in Table 7-1) there are 40 filtering and 40 rectification operations. For the delay-and-sum and Griffiths-Jim beamforming, the array processing itself provides some computational overhead (very minimal for delay-and-sum, a bit more for Griffiths-Jim), but the output of these processing schemes is a monophonic signal. When inputting this into the correlation-based processing for feature generation, the filterbank and rectification stages only need to be run once. The correlation stage is simply an auto-correlation operation on a monophonic signal to calculate the energy. This provides a savings of 1/K in processing compared to a K-element array processing solely with the correlation-based processing algorithm. If the more computationally expensive mel frequency filterbank is used instead of the Seneff filterbank, the total computational cost is even greater, regardless of the processing method chosen.

All three array processing methods in Table 7-1 require the upsampling and downsampling prior to, and following, the automatic localization algorithm used to calculate the steering delays. Therefore, the amount of computation for these steps is identical for all three methods. This provides another computational drawback of our system since we require upsampling prior to correlating to allow for non-integer sample delays between input sensors. If one knows the direction of the desired signal source, or we are willing to select a set of look directions *a priori*, then we can place hard-wired delays into lookup tables and simply choose the best one. If we do not wish to rely on hard-wired delays, the automatic localization algorithm is at least capable of being implemented on some of today's fast DSP chips.

Processing Method		CC	D/S	G-J	SPHINX
Number of Filter Channels		40	40	40	14
Number of Input Sensors		7	7	7	1
Localization	Mults/ms	7373	7373	7373	N/A
	Adds/ms	6912	6912	6912	N/A
Up/Down Sampling	Mults/ms	28560	28560	28560	N/A
	Adds/ms	28448	28448	28448	N/A
Summing	Mults/ms	N/A	16	16	N/A
	Adds/ms	N/A	96	96	N/A
Adaptive Processing	Mults/ms	N/A	N/A	12400	N/A
	Adds/ms	N/A	N/A	12480	N/A
Seneff BPF	Mults/ms	54768	7824	7824	N/A
	Adds/ms	45696	6528	6528	N/A
Mel frequency BPF	Mults/ms	82432	11776	11776	N/A
	Adds/ms	73472	10496	10496	N/A
Correlation	Mults/ms	96	16	16	N/A
	Adds/ms	90	15	15	N/A
Seneff Filter Totals	Mults/ms	90797	43789	56189	600
	Adds/ms	81146	41999	54479	580
Mel Freq. Filter Totals	Mults/ms	118461	47741	60141	600
	Adds/ms	108992	45967	58477	580

Table 7-1 : Comparison of processing methods in computational complexity. Tabulated are numbers of multiplies and adds required for the computation of a 1 msec duration of speech. The cross-correlation processing (CC) requires all input sensors to be processed separately. The delay-and-sum (D/S) and the Griffiths-Jim (G-J) beamformers first combine the input sensors into a monophonic signal which must only be filtered and rectified once within each filter channel.

7.4. Summary of Results

In this chapter we have examined the results of experiments conducted to test each of the processing stages required for our multiple-microphone correlation-based front-end algorithm. For each stage, speech recognition experiments have been carried out with the type of processing for that stage varied to explore the facets of that stage which affect the ultimate goal: improved word correct recognition accuracy.

The system configuration containing the types of processing found to give us the best speech recognition performance are as follows:

- **Array architecture.** An interleaved array structure where different array element spacings are used depending on the frequency range was found to yield superior performance. Our system uses three 7-element linear arrays, one for low frequencies, one for the midrange, and one for high frequencies, interleaved into a 15-element array.

- **Steering delays.** Since our application environment targets individual users or users who will be somewhat close in proximity to the array, we found that the desired signal source projection pattern resembles that of a point source rather than a plane wave. Also, the incoming signals may not originate on-axis to the array, so some calculation of input steering delays is necessary. Our algorithm calculates the delay between adjacent pairs of input elements and applies a correction delay by upsampling the input signal to allow delays of non-integer sample amounts to be implemented. The signals are then downsampled to the original sampling rate prior to processing.

- **Filterbanks.** Each of the multiple input signals is processed through a filterbank consisting of 40 frequency bands spaced along the input signal bandwidth. The filterbank used in the final system is the one designed by Seneff[1988] that is modelled after the filter bands within the human auditory system.

- **Rectification**. The output signal from each filter band is processed through a rectifier that does not pass any negative values. The positive values are squared. This squaring halfwave rectifier provides an expansion of the signal's dynamic range.

- **Correlation**. The outputs of the rectifiers from corresponding frequency bands from each input signal are multiplied together and integrated over a 16-ms frame to produce an energy value for that frequency band.

- **Spectral-to-Cepstral features**. The 40 spectral energy features are then converted to 12 cepstral features via the discrete cosine transform (DCT). These 40 features are then used for training and testing in the automatic speech recognition system.

Two other array processing algorithms were evaluated as well in this chapter. The delay-and-sum beamformer and a traditional LMS based array processing algorithm, the Griffiths-and-Jim algorithm were compared to our correlation-based algorithm. The delay-and-sum beamformer produced recognition results that were almost as high as the correlation-based algorithm. The Griffiths-and-Jim algorithm seemed to work well with purely additive noise, but failed in the more complex (and realistic) environments where linear filtering and correlated noise sources were present.

The computational cost of the correlation-based processing is high. The need to process each input microphone through the filterbank is very costly, whereas in the delay-and-sum and Griffiths-Jim beamforming algorithms, only one pass through the filterbank is necessary. If the delay-and-sum and Griffiths-Jim algorithms are input into a much less computationally expensive monophonic processing algorithm such as LPC, then their computational cost drops even further. If steering delays can be implemented without upsampling, a large amount of computation could be saved for all of the processing methods.

Chapter 8. Discussion

In this chapter we will review and discuss some of the results that were obtained in Chapter 7. We comment on some of the possible reasons for the disparities in performance between the results of the pilot experiments and the actual recognition experiments using natural environments. We also include some suggestions for future research.

8.1. Review of Major Findings

8.1.1. Number of Microphones

Increasing the number of input microphones does increase the word recognition accuracy to a point when microphone arrays are used as part of the initial signal processing to an automatic speech recognition system. We say “to a point” because we found that the improvement appears to level off at about 8 sensors. This suggests that there may be other factors involved in word errors that the type of processing provided by microphone arrays cannot correct. In our experiments it is fair to say that the amount of extra computation and hardware needed to process additional microphones beyond 4-8 microphones appears to exceed the benefits that one obtains from including them. Nevertheless, we did find that the use of simple microphone arrays incorporating either our cross-correlation-based processing or traditional delay-and-sum beamforming does provide an improvement in word recognition accuracy over that obtained from a single-microphone system.

8.1.2. Microphone Spacing

Experiments were also performed to examine the effects of microphone spacing on the word recognition accuracy. While spatial aliasing will take place at frequencies with a wavelength of less than twice the distance between adjacent microphones, it is not known how much these aliased components affect the actual performance results, at least at higher frequencies. There is a trade-off between microphone spacing and resolution. By making the spacing smaller, the frequency at which spatial aliasing occurs increases. On the other hand, decreasing the spacing also decreases the phase difference between signals arriving at adjacent microphones, and consequently decreases resolution.

We found that in general, for a simple linear array configuration, making the spacing too small or too large degraded the performance. We ran experiments with 7 input sensors spaced at 4, 8, and

16 cm, and we found that the results were best for the 8-cm case. Similarly, in comparisons of recognition accuracy using a 7-sensor array with microphones spaced at 3, 6, and 12 cm, the 6-cm case yielded the best result. Therefore, if one were to choose a fixed spacing for a linear array for an automatic speech recognition system, we would suggest choosing a spacing in the neighborhood of 6-8 cm.

As Flanagan *et al.* [1985] first suggested, improved performance can be obtained by using interleaved arrays. In our experiments, better word recognition accuracy was obtained for an array which used three interleaved arrays of different spacings over the performance obtained by using any of the three spacings individually. The feature set for the interleaved array was formed by extracting the energy values for low frequency bands from a sub-array with the widest spacing, energy values for the high frequency bands from more narrowly spaced arrays, and mid-band energy values from a spacing between the two. We didn't experiment with interleaving more than three spacings (for reasons of limited input signal acquisition capability), but doing so would be an interesting further experiment.

The proximity of the desired speaker to the microphone array must also be considered. We compared recognition error rates obtained using two distances from the array, one meter and three meters, with SNR decreasing as the distance between the speaker and the array increases. While recognition accuracy decreased as the distance between the speaker and the array increased, the array provided improved recognition accuracy for both distances considered. In general, we believe that the use of an array processing algorithm for speech recognition systems provides benefit regardless of the proximity of the speaker to the array.

8.1.3. Localization

The initial experiments we performed assumed that the desired signal was on axis (*i.e.* along the perpendicular bisector to the linear array), so no automatic localization was employed. This was a legitimate assumption for our tests, but for a practical system a mechanism needs to be incorporated into the system that determines the location of the desired signal.

As the overall distance between the two outermost sensors of the array increases and/or the distance of the desired speaker from the array decreases, an on-axis incoming signal will experience a greater time delay to the outer sensors of the array compared to the inner sensors. We found that recognition accuracy could be improved by 6% overall (in one specific experiment) by taking this additional delay into account and compensating for it with steering delays.

Dependable and accurate source localization may be more important for the cross-correlation processing than for delay-and-sum processing. In cross-correlation processing, the processed/aligned signals are combined via a multiplicative process (the correlation) whereas for delay-and-sum they are added. The multiplication of slightly mis-aligned signals may result in a greater decrease in signal quality than the summing of the same signals.

Our automatic localization is accomplished by cross-correlating entire input signal frames between adjacent sensors and searching for the peak of the cross-correlation function. This can be done prior to running the signal through the filterbank. If the position of the desired signal is such that its arrival time to adjacent microphones falls between sample periods, then upsampling is necessary to locate the peak of the signal cross-correlation function. This requires more computation, but it is necessary because of the sub-sample delays incurred by signals between sensors in systems with short signal travel paths (close sensors and reverberant rooms).

8.1.4. Filterbank Type

We found that the performance of the cross-correlation based processing was not significantly affected by the shape of the filters in the peripheral bandpass filterbank, at least for the shapes that we considered. It is the author's opinion that the frequency isolation obtained by simply using a filterbank is more important than the actual shape of the filters within the bank.

It is important to have filters with sufficiently narrow bandwidths such that no two harmonic components of the input signal lie within the same filter band. This helps to avoid the problem of spectral smearing (sum and difference frequency components) associated with multiplying frequencies together. It is also necessary that the transition bands be sufficiently sharp and that the stop band attenuation be sufficiently great for the filters in the filterbank such that components outside the desired frequency band are attenuated to a point that their contribution to spectral smearing is not a problem.

8.1.5. Rectifiers

Experiments examining the method of rectification were performed, both as part of our initial experiments with a small number of sensors and using the more elaborate array in our later work.

It was determined that the halfwave square-law rectifier provided the best performance in both cases, regardless of the number of input microphones. This rectifier sets the signal equal to zero for negative values of the signal waveform, and squares the signal for positive values of the signal waveform. Factors which may affect the performance of a particular rectifier include the SNR of the input signal, dynamic range of the input signal, number of input microphones, etc. It was found that normalization of the input signals to a standard value using an automatic gain control did not yield any significant difference in choice of rectifier compared to not normalizing the input amplitude.

We hypothesize that the halfwave square-law rectifier is helpful because it provides a moderate (but not excessive) degree of signal emphasis to the positive portions of the signal. Using compressive rectifiers such as the Seneff rectifier yielded worse word recognition accuracy.

8.1.6. Feature Vector

The only further exploration we performed into the set of features being used was in the number of cepstral features in our feature set. We initially used the inverse cosine transform to compute 40 cepstral values for features from a set of 40 spectral energy values, one from each frequency band in our filterbank. We reduced the size of the cepstral feature vector to 12, and found no significant degradation in recognition accuracy. Further attempts to reduce the feature set were not performed only because of limitations in the software used to implement the recognizer.

8.1.7. Comparison to Other Array Processing Methods

In order to evaluate the relative success of our efforts, we compared the recognition accuracy of our best implementation of the multi-channel cross-correlation processing combined with the SPHINX I speech recognition system against that of other types of single-microphone and multi-microphone processing algorithms. Specifically, we compared our correlation-based processing algorithm with results obtained from the best implementations available of conventional delay-and-sum beamforming and traditional array algorithms such as the Griffith-and-Jim algorithm (in conjunction with our standard SPHINX-I system). We also compared our processing to that of using a single array element processed using an environmental normalization algorithm (CDCN), and we applied CDCN to the output of the correlation-based processing as well, to determine if any further improvement could be gained from environmental normalization.

In general, performance using the algorithm presented in this thesis was almost always better than performance obtained by using delay-and-sum or Griffith-and-Jim array processing algorithms utilizing the same number of elements. However, in comparison with the delay-and-sum algorithm, the improvement in recognition accuracy using our cross-correlation processing became smaller as the number of sensors used increased.

Our cross-correlation processing provided better recognition accuracy than was obtained using a single array element processed with CDCN. By applying CDCN to the array processing algorithm, additional gain in the array performance was obtained, thus re-enforcing our original finding that the benefits of array processing are complementary to those of environmental compensation algorithms such as CDCN.

8.2. Shortcomings of the System in Real Environments

We did not actually obtain as much improvement in recognition accuracy using the cross-correlation processing compared to delay-and-sum beamforming in real speech recognition experiments as we had expected based on the results of our original pilot experiments. We discuss in this section some of the possible reasons for the differences between expected and observed performance.

8.2.1. Uncorrelated Additive Noise

The pilot experiments only used artificially-added white noise that was independent of the desired speech source. Input signals in natural environments are combinations of the desired signal, stationary noises, linear filtering and reverberation effects, as well as other noise sources that are correlated with the speech source.

8.2.2. Number of Noise Sources

We also assumed in the pilot experiments that the off-axis noise arrives from only one direction. In natural environments, there are multiple sources of off-axis noise, including multiple reflections of the noise sources due to the reverberant nature of the room.

8.2.3. Poor Localization

Especially in low SNR cases, it is difficult to localize the desired signal. The multiplicative nature of the algorithm is likely more sensitive to poor localization than other algorithms (notably delay-and-sum). In the pilot experiments, all input signals were the exact same signal, so perfect localization was guaranteed.

8.2.4. Off-Axis Noise Delay

The greatest differences in performance in the pilot experiments between the cross-correlation system and the delay-and-sum system were observed using artificial signals in which the interference source arrived at the sensors with large phase differences. This suggests that the recognition accuracy of the real system might have been greater if signals arrived at the sensors with greater spatial resolution. This could be accomplished by increasing the actual sensor spacing or by using greater amounts of computationally-costly upsampling.

8.2.5. Performance Metric

We have evaluated the cross-correlation algorithm in terms of speech recognition accuracy only. The pilot experiments used different figures of merit that were based on preservation of spectral contours. Many factors impact on speech recognition performance. Better SNR and/or better human intelligibility of speech don't necessarily correlate with improved computer speech recognizer accuracy. It is also possible that we have reached the limit of gain in recognition accuracy that this type of processing can provide for this database.

8.3. Major Contributions of this Work

This thesis is one of the first studies of the use of microphone arrays to improve the accuracy of automatic speech recognition systems. We consider the major contributions of this work to include the following:

- We introduced a new multi-microphone algorithm for enhancing spectral and cepstral features for speech recognition in natural environments. The algorithm is based on cross-correlation processing of the microphone outputs after bandpass filtering and rectification. It is motivated by our knowledge of human spatial perception.
- We analyzed the components of the cross-correlation processing system, with the goal of maximizing speech recognition accuracy.
- We analyzed the performance of the system in several natural environments. We compared recognition accuracy in detail to that obtained using delay-and-sum beamforming and to a lesser extent using the traditional Griffiths-Jim algorithm. We found that the correlation-based processing provided improved recognition accuracy over a monophonic system in real environments and performed slightly better than traditional delay-and-sum beamforming in

the same environments. The Griffiths-Jim beamformer did not perform well in real environments.

- We demonstrated both with the cross-correlation system and with delay-and-sum beamforming that the benefits provided by array processing are complementary to those provided by acoustical pre-processing algorithms such as codeword-dependent cepstral normalization (CDCN).
- We confirmed that speech recognition accuracy can be improved by using a compound array with sub-arrays with different element spacing, as had been proposed by Flanagan *et al.* [1985].

8.4. Suggestions for Future Work

In this final chapter, we provide some areas of future research directions for our multi-microphone cross-correlation based front end processing.

8.4.1. Filterbank Representation in Hardware

Currently, the computational requirements of our system are extremely large, and make choosing our processing algorithm over a less computationally expensive algorithm such as delay-and-sum beamforming unlikely. Computation could be speeded either by implementing the filterbank section of the algorithm on a parallel machine, or by implementing the filters in hardware using a fast digital signal processor to perform the computation.

8.4.2. Better Localization

Since our research interest was not geared toward the localization aspects of the array, more could be done to analyze the effect of localization on the performance of the algorithm. Higher up-sampling rates could be explored and more exact localization algorithms could be applied to the input signals prior to processing.

8.4.3. Addition of Inhibition Mechanisms

As mentioned previously, the human auditory system exhibits a phenomenon known as the “precedence effect”. Inhibition mechanisms such as those proposed by Lindemann [1986] could be added to the processing. Lindemann has demonstrated success in using auditory models with inhibition to aid in localization. An inhibition model placed after the rectification stages may help in reverberant cases by applying some modeling akin to the precedent effect in our human auditory system. Applying an inhibition model prior to the filterbanks may be a better localization mechanism than our current localization algorithm. Bodden and Blauert [1992] have done experiments demonstrating improvement in handling the "cocktail party" effect by imbedding some of Lindemann's inhibition ideas into their systems. More recently, Bodden and Anderson [1995] demonstrated an increase in phoneme recognition accuracy using a similar system.

One drawback to adding inhibition is that it would slow down the processing time further because the inhibition structure would have to be applied to each filter output in the system.

8.4.4. Integration Method

Currently the correlation operation is performed for a single frame length by multiplying filtered and rectified frames within a frequency band from each input signal, then summing the values together over the frame length. One might impose a form of temporal integration by averaging current and past frame spectral energy values within a frequency band to use as the feature set. By applying a weighting function (for example, a decaying exponential over time) to the averaging process, one could imply an associated time constant to the averaging that would favor the most recent energy values and decrease the contribution of past values.

8.4.5. Comparison to Other Adaptive Systems

Further comparisons to other array processing algorithms and adaptive systems could be made as well. Our current system has no dynamic adaptation capability other than what is built in to the localization component. Imbedding adaptation to incoming noise or environmental conditions may be possible, primarily at the filtering and correlation stages.

8.4.6. Further Work on Array Element Placement

All of the experiments carried out in this thesis used a one-dimensional (horizontal) linear placement of array elements. Even the interleaved arrays were really superposition of three linear bandlimited arrays. The geometry of the array could be varied to see how the algorithm performs with different microphone placement. Adding vertical elements might be interesting, placing elements around the screen of the computer, trying non-linear spacing, etc. At AT&T, Flanagan *et al.* [1985] had a two-dimensional delay-and-sum array built for one of the company's lecture halls, and more recently, Flanagan [1992] has proposed three-dimensional systems placed on chandeliers in rooms, etc.

8.5. Summary

In this research we proposed and evaluated a multi-microphone system that is based on the cross-correlation processing of the human auditory system. The cross-correlation-based system was motivated by a desire to achieve a gain in recognition accuracy that is greater than what is provided by direct delay-and-sum beamforming, and in a fashion that is less sensitive to reverberation effects than traditional adaptive filtering approaches.

The validity of the design principles was confirmed in a series of pilot experiments which demonstrated that the cross-correlation processing can maintain the spectral contours of speech samples at lower SNRs to a greater extent than delay-and-sum beamforming.

The speech recognition accuracy provided by the cross-correlation system was evaluated by a series of experiments measuring recognition accuracy for speech in a noisy computer laboratory and in a less noisy but more reverberant conference room. It was found that the cross-correlation-based processing provided best performance in virtually every case considered, but the margin of improvement in recognition accuracy relative to delay-and-sum beamforming was frequently smaller than expected. Our expectations were confirmed that traditional adaptive array processing was not effective in natural environments, at least for the implementations considered.

We also confirmed that the benefits in recognition accuracy derived from array processing are complementary to those provided by noise compensation algorithms such as codeword-dependent cepstral normalization (CDCN).

Bibliography

Acero, A.; "Acoustic and Environmental Robustness in Automatic Speech Recognition", Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, September 1990.

Acero, A. and Stern, R. M.; "Environmental Robustness in Automatic Speech Recognition", ICASSP-90, pp. 849-852, April 1990.

Allen, J. B., Berkley, D. A., and Blauert, J.; "Multi-microphone Signal Processing Technique to Remove Room Reverberation From Speech Signals", JASA, 62, pp. 912-915, 1979.

Berouti, M., Schwartz, R., and Makhoul, J.; "Enhancement of Speech Corrupted by Acoustic Noise", ICASSP-79, pp. 208-211, April 1979.

Berthommier, J., Holdsworth, J., Schwartz, J. L., and Patterson, R.; "A Multi-representation Model for Auditory Processing of Sounds", Auditory Physiology and Perception-9th International Symposium on Hearing, 1991.

Blauert, J.; "Psychoacoustic Binaural Phenomena", Proceedings of the 6th International Symposium on Hearing. Bad Nauheim, Germany, April 5-9, 1983.

Bloom, P. J.; "Evaluation of a Dereverberation Process by Normal and Impaired Listeners", ICASSP-80, pp. 500-503, April 1980.

Bodden, M. and Blauert, J.; "Separation of Concurrent Speech Signals: A Cocktail-Party -Processor for Speech Enhancement", ESCA-Workshop on Speech Processing In Adverse Conditions, pp. 1-4, Cannes, France, 10-13 November 1992.

Bodden, M. and Anderson, T. R.; "A Binaural Selectivity Model for Speech Recognition", Eurospeech '95, Madrid, Spain, September 1995.

Boll, S. F.; "Suppression of Acoustic Noise in Speech Using Spectral Subtraction", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 27, pp. 113-200, 1979.

Boll, S. F., and Pulsipher, D. C.; "Suppression of Acoustic Noise in Speech Using Two Microphone Adaptive Noise Cancellation", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28, pp. 752-753. August 1980.

Crochiere, R. E. and Rabiner, L. R.; "Multirate Digital Signal Processing", Prentice-Hall Signal Processing Series, pp. 93-97, 1983.

Davis, S. B., and Mermelstein, P.; "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 28, pp. 357-366, August 1980.

Flanagan, J. L., Johnston, J. D. Zahn, R., and Elko, G. W.; "Computer-Steered Microphone Arrays for Sound Transduction in Large Rooms", JASA, Vol. 78, pp. 1508-1518, November 1985.

Flanagan, J. L.; Research at the CAIP Center, Rutgers University. Unpublished Documents from the 1992 Workshop On Microphone Arrays: Theory, Design, and Application.Brown University, Providence, RI, October 1992.

Frost, O. L.; “An Algorithm for Linear Constrained Adaptive Beamforming”, Proceedings of IEEE Vol. 60, pp. 926-935, 1972.

Ghitza, O.; “Robustness Against Noise: The Role of Timing-Synchrony Measurement”, ICASSP-87, pp. 2372-2375, April 1987.

Griffiths, L. J. and Jim, C. W.; “An Alternative Approach to Linearly Constrained Adaptive Beamforming”, IEEE Transactions on Antennas and Propagation, AP-30(1), pp. 27-34, Jan. 1982.

Hikichi, T. and Itakura, F.; “Time Variation of Room Acoustic Transfer Functions and its Effects on a Multi-Microphone Dereverberation Approach”, ATR Technical Report, 1994.

Huang, X.-D.; “The SPHINX-II Speech Recognition System: An Overview”, Computer Speech and Language, Vol. 2, 1993.

Jeffress, L. A., “A Place Theory of Sound Localization,” Journal of Computational Physiology and Psychology, Vol. 61, pp. 468-486, 1948.

Juang, B. -H.; “Speech Recognition in Adverse Environments”, Computer Speech and Language, Vol. 5, pp. 275-294, July 1991.

Kellerman, W.; “A Self-Steering Digital Microphone Array”, ICASSP-91, pp. 3581-3584, April 1991.

Lee, K.-F.; “Automatic Speech Recognition -- The Development of the SPHINX System”, Kluwer Academic Publishers, 1989.

Lindemann, W.; “Extension of a Binaural Cross-Correlation Model by Contralateral Inhibition. I. Simulation of Lateralization for Stationary Signals.”, JASA, Vol. 80, pp. 1608-1622, December 1986.

Lindemann, W.; “Extension of a Binaural Cross-Correlation Model by Contralateral Inhibition. II. The Law of the First Wave Front.”, JASA, Vol. 80, pp. 1623-1630, December 1986.

Liu, F.-H.; “Environmental Adaptation or Robust Speech Recognition”, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, May 1994.

Lyon, R. F.; “A Computational Model of Binaural Localization and Separation”, ICASSP-83, pp. 1148-1151, April 1983

Moreno, P. J.; “Robust Algorithms for Automatic Speech Recognition”, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, April 1996.

Moreno, P. J., Raj, B., and Stern, R. M.; “Approaches to Environment Compensation in Automatic Speech Recognition”, International Conference on Acoustics-95, Trondheim, Norway, 1995.

Morii, S.; “Spectral Subtraction in the SPHINX System”, Unpublished work at Carnegie Mellon University. 1988.

Ohshima, Y.; “Environmental Robustness in Speech Recognition using Physiologically-Motivated Signal Processing”, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, December 1993.

Pallett, D. S., Fiscus, J. G., Fisher, W. M., Garofolo, J. S., Martin, A. F., and Przybocki, M. A.; "1995 HUB-3 NIST Multiple Microphone Corpus Benchmark Tests", Conference Proceedings from the ARPA Speech Recognition Workshop, February 18-21, 1996.

Palm, S. R.; "Enhancement of Reverberated Speech", M. S. Thesis, Carnegie Mellon University, Pittsburgh, PA., May 1989.

Parks, T. W., and McClellan, J. H.; "A Program for the Design of Linear Phase Finite Impulse Response Filters", IEEE Transactions on Audio Electroacoustics, Vol. AU-20, No. 3, pp. 195-199, August 1972.

Peterson, P. M.; "Adaptive Array Processing for Multiple Microphone Hearing Aids", RLE Technical Report No. 541. Research Laboratory of Electronics, MIT, Cambridge, MA. February 1989

Picone, J.; "Continuous Speech Recognition Using Hidden Markov Models", IEEE ASSP Magazine. Vol. 7(3):pp. 26-41, July 1990.

Rabiner, L. R. and Juang, B. -H.;"An Introduction to Hidden Markov Models", IEEE ASSP Magazine, Vol. 3 (1): pp 4-16. January 1986.

Rabiner, L. R. and Juang, B. -H.; "Fundamentals of Speech Recognition", Prentice-Hall Signal Processing Series, 1994.

Rabiner, L. R. and Schafer, R. W.; "Digital Processing of Speech Signals", Prentice-Hall Signal Processing Series, 1978.

Schafer, R. W.; "Echo Removal by Discrete Generalized Linear Filtering, Tech Report 466, MIT Research Laboratory of Electronics, MIT, Cambridge, MA, February 1969.

Seneff, S.; "A Joint Synchrony/Mean-Rate Model of Auditory Speech Processing", Journal of Phonetics, Vol. 16, No. 1, January 1988, pp. 55-76.

Silverman, H. F., Kirtman, S. E., Adcock, J. E. and Meuse, P. C.; "Experimental Results for Baseline Speech Recognition Performance using Input Acquired from a Linear Microphone Array", Brown University Technical Report, Brown University, Providence, RI, 1992

Sondhi, M. M. and Berkley, D. A.; "Silencing Echoes on the Telephone Network", Proceedings of the IEEE, Vol. 68, No. 8, pp. 948-963, August 1980.

Stern, R. M. and Colburn, H. S.; "Theory of Binaural Interaction Based on Auditory-Nerve Data. IV. A model for Subjective Lateral Position", JASA, Vol. 64, pp. 127-140, 1978.

Stockham, T. G., Cannon, T. M., and Ingebretsen, R. B.; "Blind Deconvolution Through Digital Signal Processing", Proceedings of the IEEE, Vol. 63, pp. 678-692, 1975.

Widrow, B., Glover, R., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., Zeidleer, J. R., Dong, E., and Goodlin, R. C.; "Adaptive Noise Cancelling: Principles and Applications", Proceedings of IEEE 63, pp 1692-1716, December 1975.

Widrow, B. and Stearns, S. D.; "Adaptive Signal Processing", Prentice-Hall, Englewood Cliffs, NJ, 1985.

Van Compernolle, D.; "Switching Adaptive Filters for Enhancing Noisy and Reverberant Speech from Microphone Array Recordings", ICASSP-90, pp. 833-836, April 1990.

Vea, M.; "Multisensor Signal Enhancement for Speech Recognition", M.S. Thesis, Carnegie Mellon University, Pittsburgh, PA, September 1987.

Yamada, H., Wang, H., and Itakura, F.; "Recovering of Broadband Reverberant Speech Signals by Sub-band MINT Method", ICASSP-91, pp. 969-972, April 1991.

Zelenski, R.; "A Microphone Array with Adaptive Post-Filtering for Noise Reduction in Reverberant Rooms.", ICASSP-88, pp. 2578-2581, April 1988.

Zurek, P. M.; "The Precedence Effect and its Possible Role in the Avoidance of Interaural Ambiguities", JASA, 67(3), pp. 952-964, 1980.

Appendix A. Signal Processing

In this Appendix, we provide the equations necessary to implement the processing units of the correlation-based processing introduced in this thesis. Some of the processing has been outlined in previous chapters, and some has been implemented directly from the literature. In the latter cases, references will be provided rather than reprinting the equations here.

A.1. The Input Processing: Upsampling, Localization, and Downsampling

A set of K input sensors was used as inputs into a multi-channel A/D converter to collect the speech input data for our experiments. The collected input signals, $s_k[n]$, where k is the sensor number and n is the time index (the total number of samples in the utterance is N), are first unbiased by subtracting the mean of each signal from each sample within the signal.

$$x_k[n] = s_k[n] - \frac{1}{N}\sum_{n=0}^{N-1} s_k[n]$$

These K input sensors were then time-aligned in the direction of the desired signal by first upsampling all of the signals by a factor of R (R equals 8 in this thesis), applying an automatic localization algorithm to extract the time delay between the sensors and correct for it to align the signals, then downsampling the signals (again by a factor of 8) to return the signals to their original sampling rate. (The reasons for the upsampling are discussed in Section 7.1.3.) The desired signal in any sensor was assumed to have a greater signal power than the noise or undesired signals. This assumption allows us to choose the location of the peak along the time axis of the cross-correlation function between any two input sensors as the value of the time delay for the desired signal between those two sensors.

A.1.1. Upsampling and Downsampling

The upsampling of the input signals prior to calculating the steering delays, and the downsampling of the time-aligned signals after applying the steering delays can be accomplished by any standard interpolation and decimation sampling routine. We chose to use an FIR filter-based approach because of the linear phase nature of FIR filters. Because of the computation required in upsampling, we chose to use the *polyphase filtering* method discussed in Crochiere and Rabiner [1983] (see the reference for implementation details). In this method, an anti-aliasing FIR filter of length L, designed by the Parks-McClellan algorithm [Parks and McClellan, 1972], is applied during the upsampling operation by dividing the filter into R sub-filters with lengths L/R.

The downsampling operation which is applied after the auto-localization processing simply involves selecting every R^{th} sample from the processed, upsampled signals.

A.1.2. Auto-Localization

After the signals have been upsampled, we are left with a set of K upsampled input signals of the form: $x_k[n]$, where k is the microphone sensor number and n is the upsampled time (the number of samples in the input signals is now RN). These signals are then input to the auto-localization algorithm.

Localization is performed on an utterance-by-utterance basis in our system, the assumption being that the subject does not change position substantially over the length of an utterance. We wish to use for localization only the frames in the utterance where the desired speech signal is present. Since the SNR of each input signal is also assumed to be approximately equal, we can select any one input signal and use that signal to select the frames for localization. Therefore, we choose $x_1[n]$ to be the input signal used to identify the localization frames.

Let the frame length be W. The energy, E_i, is calculated over the length of a frame, then stepped by a frame length to begin calculating the energy of the next frame, where i is the integer frame index beginning at 0. There is no frame overlap for the energy calculations at the localization stage, unlike later portions of the processing.

$$E_i = \frac{1}{W}\sum_{l=0}^{W-1} (x_1[l+iW])^2$$

The log energy is calculated by taking the log of E_i. After all of the frame energy values have been calculated, we find the maximum frame energy, E_{max}, and the minimum frame energy, E_{min} from the set of calculated frame energies. The value of the frame energy threshold above which we say a frame contains the desired signal is then

$$Thr = E_{min} + \alpha(E_{max} - E_{min})$$

where a is between 0 and 1. For the localization calculations, we set $a = 0.85$. This gave us somewhere between 10-20% of the frames of the input signals to be used in the localization algorithm to determine the values of the steering delays.

The next step is to determine the steering delay between adjacent pairs of sensors, $x_k[n]$ and $x_{k+1}[n]$. This is done by calculating the cross-correlation function, $c_{k,k+1}[m]$ for the pair within the selected energy frames. We need only calculate the function for a small range ($-M <= m <= M$) around the center ($m=0$), as the geometry of the sensor placement in the array will only allow a certain range of possible locations for the peak of the function due to spatial aliasing constraints (see Section 4.1.3). M was chosen to be a maximum of 5 samples prior to upsampling by a factor of $R=8$, for a maximum of $M=40$ for the upsampled signals.

$$c_{k,\,k+1}[m] = \frac{1}{2M+1}\sum_{n=-M}^{M} x_k[m]\,x_{k+1}[m+n]$$

The location of the maximum peak of the cross-correlation function is identified for each selected energy frame, and then these peak locations are averaged across all of the selected energy frames to yield the final steering delay, d_k, between that adjacent pair of sensors. The delay times for the entire set of K sensors, $m_{1,k}$ (from Figure 5-1), are then formed by summing the values of d_k (for the $K-1$ adjacent pairs of sensors) up to that point, with $m_{1,1} = 0$.

$$m_{1,\,k} = \sum_{j=1}^{k-1} d_j$$

The steering delay values are then applied to the input signals to time-align them and the signals are downsampled back to the original input sampling rate.

A.2. Correlation-Based Algorithm Processing

The correlation-based algorithm was introduced and discussed in Chapter 5. The filterbank implementation for the Seneff filterbank is discussed in Seneff[1988]. The Mel-frequency filters are described in Davis and Mermelstein [1980]. Equations are provided in Chapter 5 for the cross-correlation operation and the DCT conversion for the spectral features into cepstral features.

A.3. Delay-and-Sum Beamformer Processing

The delay-and-sum beamforming is implemented post-time-aligning and prior to any filterbank operations. The operation involves forming a monophonic input signal by summing all of the time-aligned input signals and averaging the sum over the number of input signals.

$$x'[n] = \frac{1}{K}\sum_{k=1}^{K} x_k[n]$$

The monophonic signal $x'[n]$ is then processed through the chosen filterbank and rectifier to yield a set of C output signals $y[n,\omega_c]$, where c is the index of the filter channel. The correlation operation to obtain the output spectral energy, E_c, for this monophonic signal is now an auto-correlation over the frame length, N, as opposed to the cross-correlation when using multiple input signals.

$$E_c = \sum_{n=0}^{N-1} (y[n,\omega_c])^2$$

The DCT operation to convert the spectral energy values, E_c, to cepstral coefficients is the same as in Section 5.1.

A.4. Griffiths-Jim Beamformer Processing

The Griffiths-Jim beamformer was introduced in Section 2.2 and is shown in Figure 2-5. It consists of the summing process from the delay-and-sum beamformer, to provide an enhanced desired signal component, and an adaptive multi-channel section which uses the differences between the time-aligned input signals to provide a representation of the undesired signal components (see Widrow and Stearns [1985]). If the desired signal is time-aligned and equivalent in signal strength in all of the input signals, then taking the difference of adjacent input signals will cancel the desired component completely, leaving only a signal representing the undesired signal components contained in the two signals. If we then filter and combine the undesired components from all of the difference pairs with a set of filters that has had their weights adaptively adjusted to minimize the error between the enhanced desired signal component and the filtered difference signals, we can cancel much of the undesired signal.

The output of the Griffiths-Jim beamformer is a monophonic signal which is then presented to the correlation-based processing exactly as the monophonic delay-and-sum signal was in Section

A.3. If the signal components in each of the time-aligned input signals are not exactly equal, some of the desired signal component will be present in the difference signals. Adapting the filter weights while these signals are present would lead to signal cancellation problems, as discussed in Section 2.2, so we only adapt the filter weights during portions of the input signals where the desired signal component is not present. The filter weights are frozen during frames where the desired signal is present.

The portions of the input signals where the desired signal is not present are determined from the frame energy in the same way they were determined for the localization processing in Section A.1.2. The maximum and minimum frame energies, E_{max} and E_{min}, are calculated for the entire utterance, as before. An energy threshold is now set *below* which it is determined that no desired signal is present in that frame:

$$Thr = E_{min} + \beta (E_{max} - E_{min})$$

where β is now a parameter value between 0 and 1. β was chosen to be 0.75 for the Griffiths-Jim threshold. For frames with energy below Thr, the filter taps will be updated; they will be frozen if the frame energy is above Thr.

The summing signal, $x'[n]$, is formed exactly as it was in the delay-and-sum beamformer:

$$x'[n] = \frac{1}{K} \sum_{k=1}^{K} x_k[n]$$

The $K\text{-}1$ difference signals, $x_{d,k}[n]$, are formed by subtracting adjacent pairs of input signals:

$$x_{d,k}[n] = \frac{1}{2} (x_{k+1}[n] - x_k[n])$$

The difference signals are then convolved with their respective filter weights, $w_k[n]$, and the output signals are summed to yield the undesired signal, $x_d[n]$:

$$x_d[n] = \sum_{k=1}^{K-1} (x_{d,k}[n] \otimes w_k[n])$$

The monophonic output signal, $x_{gj}[n]$, is then formed by subtracting the undesired signal from the summed enhanced signal:

$$x_{gj}[n] = x'[n] - x_d[n]$$

It is this signal, $x_{gj}[n]$, which will be input to the filterbank during the correlation-based processing.

The filter taps are updated by adapting the weights, $w[n]$, as described in Widrow and Stearns [1985].