Acoustic Beamforming Using Microphone Arrays

by

Robert Jack Lustberg

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1993

Author .....
Department of Electrical Engineering and Computer Science
May 10, 1993

Certified by .....
William Rabinowitz
Principal Research Scientist
Thesis Supervisor

Certified by .....
Gary Elko
Member of Technical Staff, AT&T Bell Laboratories
Thesis Supervisor

Accepted by .....
Campbell Searle
Chairman, Committee on Graduate Students

Acoustic Beamforming Using Microphone Arrays

by

Robert Jack Lustberg

Submitted to the
Department of Electrical Engineering and Computer Science

May 10, 1993

In Partial Fulfillment of the Requirements for the degrees of
Bachelor of Science
and
Master of Science

Abstract

A digital implementation of a wideband steerable acoustic beamformer has been designed and successfully built. The hardware implementation of the acoustic beamformer uses a twenty-nine element harmonically nested microphone array, associated preamplification and sampling hardware, six AT&T DSP3210 processors, and a personal computer. The acoustic beamformer bandwidth is between 250Hz - 7200Hz. Beam steering is provided by forming seven beams at predetermined angles and then selecting one or more beams to be on. Beam selection can be done manually or automatically. Automatic beam selection uses a simple voting algorithm whose main parameter is the detected energy in each of the seven beams. Furthermore, the digital acoustic beamformer can be easily reconfigured to provide constant beamwidth beamforming (as a function of both steering angle and frequency), variable beamwidth, and rapid beamwidth modulation.

Thesis Supervisor: William Rabinowitz
Title: Principal Research Scientist

Thesis Supervisor: Gary Elko
Title: Member of Technical Staff, AT&T Bell Laboratories

Acknowledgements

I would like to dedicate this thesis, the final fruition of my five years at MIT, to my parents and brother. It is thanks to their support that I have been able to attend and enjoy these years in Boston. I would also like to thank my brother for going to Stanford, giving me an excuse to get out of the Northeast and go to California a few times. Anjalika, thanks for making the last three years an incredible time. I would also like to thank all my friends at MIT for making Boston a most excellent place to hang-out.

Gary, the ultimate thesis supervisor, always providing new ideas and different perspectives, thanks for everything. I'd also like to thank William Rabinowitz for supervising my thesis at MIT. I'd like to thank Juergen and Sunil at Bell Labs without whom I would have probably been living on the streets of New Providence. Many thanks to Syd for not killing me for taking over his workspace with my thesis. I would like to thank Bob Kubli and Keith Bauer for helping me get my work off the ground. I would finally like to thank AT&T Bell Labs for supporting the MIT VI-A program and allowing students like myself access to first-rate facilities and staff.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Within the past few decades, the globalization of the marketplace and the increased integration of information have drastically altered the environment in which companies and governments transact business. Large corporations must provide branches in different parts of the world and keep the branches linked through some effective mode of communication. Smaller companies find themselves more often in alliances with other firms which are not necessarily in the same physical location [1]. Teleconferencing helps provide the sort of communication needed to effectively integrate individuals separated by large distances.

Even though full duplex teleconferencing exists today, there are various factors which degrade system performance. Extraneous noise sources which find their way into the input microphones, room reverberation, and closed loop feedback problems are some of the causes of system degradation. The design and implementation of a wideband steerable microphone array is suggested as a means of improving full duplex teleconferencing.

The use of a steerable microphone array and a steerable loudspeaker array helps reduce the effects of the aforementioned factors of system degradation. For example, by using steerable acoustic arrays we can help break the closed-loop present in teleconferencing, and thus avoid closed loop feedback problems such as howling. Another example is using the steerable microphone array to 'filter out' extraneous noise sources. Current microphone arrays [2, 3] and loudspeaker arrays [4] have been implemented to help reduce some of

these undesireable effects, but they have introduced problems of their own associated with the array implementation and its resulting beamforming characteristics. For example, microphone array beamforming results in a radiation pattern which varies as a function of frequency and scan angle. In building a new digital steerable microphone array, some of the problems associated with microphone arrays were tackled and features have been added to make the resulting technology more attractive.

The microphone array which was built and shown to work improves performance by:

- Providing an 8kHz bandwidth.
- Providing smooth automatic beam switching using a simple voting algorithm.
- Is easily reconfigureable to provide constant beamwidth beamforming as a function of frequency and as a function of scan angle.
- Allowing variable beamwidth control.

By implementing the microphone array digitally, all changes to the beamformer can be easily accomplished by using signal processing tools and incorporating them into the assembly code running on the digital signal processors. Furthermore, a digital implementation of the microphone array provides an elegant way to normalize microphone and preamplifier mismatch.

1.2 Overview

The following gives an outline of the topics covered in this thesis:

- **Chapter 2** covers the theory behind sampled aperture design together with the overlap and links between spatial filter and temporal filter design.
- **Chapter 3** covers the overall system architecture, including the hardware and software design. It discusses the underlying algorithms that are implemented in the beamformer.

- **Chapter 4** covers the hardware and software implementation. It discusses the AT&T DSP3210 [5] processors, how the algorithms are implemented, and the sampling hardware interface with the personal computer.
- **Chapter 5** covers the experimental results obtained from using the acoustic beamformer in the manual and automatic steering modes.
- **Chapter 6** discusses the issue of constant beamwidth beamforming as a function of steering angle and frequency and rapid beamwidth modulation.
- **Chapter 7** provides a discussion of the achieved results and future directions.

Chapter 2

Sampled Aperture Theory and Beamforming

2.1 Sampled Aperture Response

The spatial response of an arbitrary array can be determined in two ways. The first method involves deriving the response from results in field theory. The second method involves deriving the response using a geometric construction and exploiting knowledge on wave propagation. Since each method provides a different perspective on the same problem, it is instructive to derive the array response using both methods.

2.1.1 Array Response: Result from Field Theory

From results in field theory [6, 7] the far-field radiation/reception pattern of a continuous aperture is defined as:

$$f(\vec{\mathbf{k}}, \vec{\mathbf{r}}) = \int a(\vec{\mathbf{r}}) e^{-j\vec{\mathbf{k}}\cdot\vec{\mathbf{r}}} d\vec{\mathbf{r}} \tag{2.1}$$

In equation 2.1 generalized vector notation has been used to indicate the wavevector ($\vec{\mathbf{k}} = \frac{\omega}{c}\vec{\mathbf{v}}$, where $\vec{\mathbf{v}} = \frac{\vec{\mathbf{k}}}{|\vec{\mathbf{k}}|}$) and the location vector ($\vec{\mathbf{r}}$). Equation 2.1 shows that the far-field radiation pattern of a continuous aperture $a(\vec{\mathbf{r}})$ is its Fourier transform in $\vec{k}$ space. Sampling $a(\vec{\mathbf{r}})$ at $M = 2N + 1$ equispaced locations, the sampled aperture frequency response is given

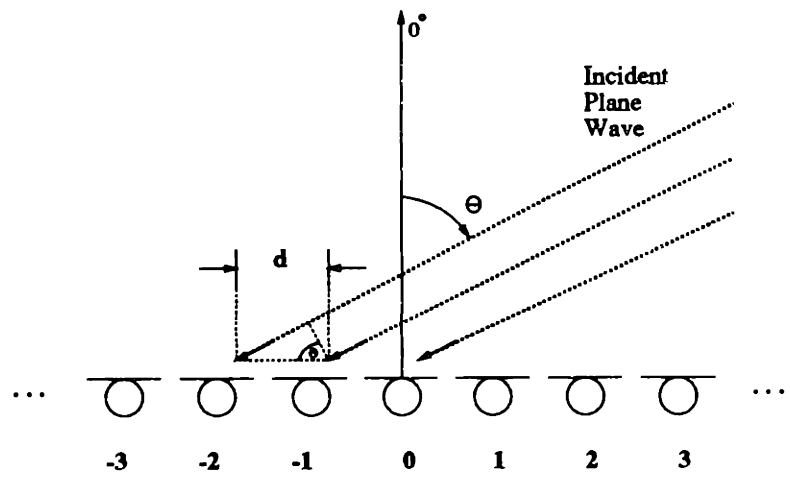

Figure 2-1: Plane wave incident on a microphone array.

by:

$$H(\vec{\mathbf{k}}, \vec{\mathbf{r}}_n) = \sum_{n=-N}^{N} a_n e^{-j\vec{\mathbf{k}}\cdot\vec{\mathbf{r}}_n} \tag{2.2}$$

where $\vec{\mathbf{k}}$ still represents the wavevector and $\vec{\mathbf{r}}_n$ represents the discrete locations of the aperture samples $\{a_n\}$. Equation 2.2 describes the Fourier transform of the aperture samples $\{a_n\}$. If the planar wavefront is propagating incident to a linear array of length $(M-1)d$ and the aperture samples are spaced at a constant interval d then:

$$H(k, u) = \sum_{n=-N}^{N} a_n e^{jkndu} \tag{2.3}$$

$$\vec{\mathbf{k}} \cdot \vec{\mathbf{r}}_n = -|k||r_n| \sin\theta = -knd \sin\theta$$

$$u = \sin\theta$$

where $|k| = \frac{\omega}{c}$. The angle θ in equation 2.3 is defined as relative to the perpendicular of the line array axis as shown in figure 2-1. One should note the indexing used in figure 2-1.

If the set of a_n were symmetric, such that $\{a_n\} = \{a_{-n}\}$, the expression for $H(k, u)$ could be written as:

$$\hat{H}(k, u) = H(k, u) = a_0 + \sum_{n=1}^{N} a_n \cos(kndu) \tag{2.4}$$

The motivation for rewriting the expression in equation 2.3 is to illustrate that with a symmetric set of weights the spatial response of the linear array is real and even.

Having used results from wave theory, an expression for the array response of a linear microphone array of length $(M-1)d$ has been derived. Arriving at the solution using this method provides insight in describing the radiation pattern as a simple spatial fourier transform.

2.1.2 Array Response: Result from Geometric Construction

A second way to derive the array response for a finite linear array is to use a geometric construction. If the elements in figure 2-1 are defined as microphones, the extra distance travelled by the wavefront from one element to the next can be defined as $d\sin\theta$. With the speed of sound in the medium equal to c, the interelement delay time τ can be defined as:

$$\tau = \frac{d\sin\theta}{c} \tag{2.5}$$

If the microphone outputs are scaled with a set of symmetric $\{a_n\}$ and the delay τ is applied to indicate the actual wavefront delay, the array response can be written as:

$$H(\omega,\theta) = \sum_{n=-N}^{N} a_n e^{\frac{-j\omega nd\sin\theta}{c}} \tag{2.6}$$

The result in equation 2.6 holds for plane wave propagation where the amplitude of the impinging wave has been normalized to one.

$H(\omega,\theta)$ in equation 2.6 is equal to $H(k,u)$ in equation 2.3. To obtain a more generalized form of the array response consider steering the radiation pattern by scaling the microphone weights $\{a_n\}$ with a progressive exponential of the form $e^{jk\alpha nd}$. Applying this exponential results in a shifted radiation pattern $H(k,u-\alpha)$, where $\alpha = \sin\theta_0$. The origin of the radiation pattern in polar coordinates would shift from $\theta = 0$ to $\theta = \theta_0$. Using this result, a more generalized microphone array response can be described with:

$$H(\theta,\theta_o,\omega) = \sum_{n=-N}^{N} a_n e^{\frac{-j\omega nd(\sin\theta-\sin\theta_o)}{c}} \tag{2.7}$$

Using geometric construction and knowledge of plane wave propagation the same result obtained in the previous section has been arrived at. Equation 2.7 provides the generalized radiation pattern for a linear microphone array with equispaced elements. With these results, the analogy between array response design and FIR filter design can be examined.

2.2 Sampled Aperture Design as FIR Filter Design

From signal processing theory [8] the Fourier transform of a uniformly sampled discrete time signal is given by:

$$H(\omega_t) = \sum_{all\ n} a[n]e^{-jn\omega_t} \tag{2.8}$$

where ω_t indicates the frequency variable in the temporal domain (should be distinguished from ω in the previous section) and $a_n = a[n]$. It is directly visible that the discrete time Fourier transform looks identical in form to the spatial Fourier transform described in equation 2.3 or 2.6 (where $u = \sin\theta$). One need only perform the mapping $\omega_t \rightarrow kdu$ to get from temporal frequency space to u space. To obtain the array response as a function of scan angle (instead of kdu) the transformation $\theta = \sin^{-1} u$ can be performed to get from u-space to θ-space.

By using the aforementioned transformation and mapping, it is easy to see the resulting spatial Fourier transform is identical to the temporal Fourier transform where the ω_t axis in the time Fourier transform has been warped by the following transformation:

$$\theta = \sin^{-1}(\frac{\omega_t}{kd}) \tag{2.9}$$

Applying the above transformation warps the ω_t axis into polar coordinates (θ). The actual amplitude of the response at the associated frequency has not changed. Since ω_t is being mapped into θ, one should note that the ω_t seen in equation 2.9 has nothing to do with the associated ω of the impinging wave. The frequency of the impinging wave is hidden in the wavenumber k. Figure 2-2 shows the result of applying equation 2.8 to an eleven element, uniformly weighted signal, and then applying the transformation shown in equation 2.9 on the ω_t axis to get the spatial response of a microphone array with uniform weights and

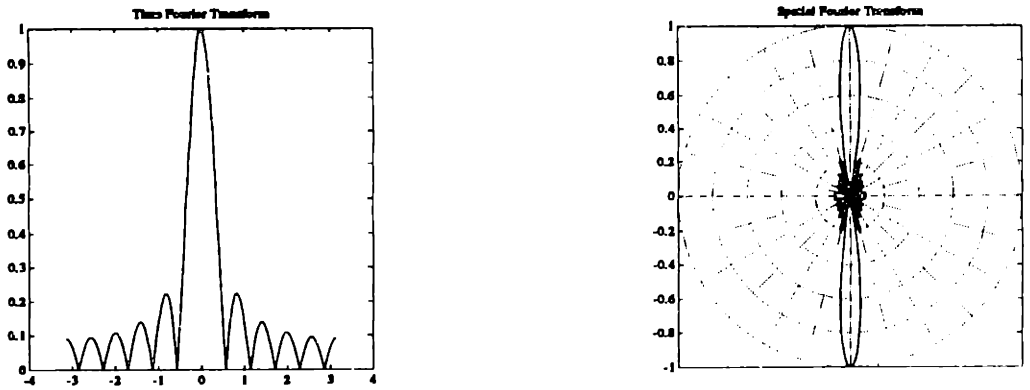

Figure 2-2: Uniformly weighted array response derived using FIR filter design

spaced at $d = \frac{\lambda}{2}$.

The above results imply that microphone array design is almost identical to FIR filter design; yet there are two issues that surface in microphone array design. The first issue is realizing the constraint that is imposed on the argument of the $\sin^{-1}$ function in equation 2.9. The argument in equation 2.9 must be less than one, which requires that $|u| \leq 1$. The visible domain of u is thus defined with the inequality $|u| \leq 1$. Understanding the constraint imposed on u is important to avoid spatial aliasing.

The second issue is noting that in equation 2.9, the ω_t variable that is being mapped into θ is scaled by the factor $kd = \frac{2\pi d}{\lambda}$. This scaling factor causes the width of the spatial filter to vary as a function of frequency.

2.3 Spatial Aliasing and Beamwidth

In the previous section the analogy between FIR filter design and microphone array design was discussed. The resulting spatial response of a microphone array has been referred to as the radiation/reception pattern of the array. The design of this spatial response is sometimes referred to as array beamforming since a spatial response is determined that closely resembles a beam (see Figure 2-2). From now on, the terms spatial filter, beam, and beamforming will be used interchangeably.

2.3.1 Spatial Aliasing

Spatial filter design, like FIR filter design, is subject to aliasing. As was noted in the previous section, the visible domain of u is defined as $|u| \leq 1$. The mapping $\omega_t \rightarrow u$ maps a cartesian coordinate ω_t into a cartesian coordinate u. Therefore, if equation 2.8 is used to obtain the transform of an eleven element, equally weighted signal, and then apply the transformation $\omega_t \rightarrow kdu$, the spatial response of this array can be plotted in terms of u. The upper left plot in figure 2-3 shows the array response of an eleven element equally weighted array where the horizontal axis is u and kd has been set to π.

As in the case of FIR filter design, the response is replicated outside the visible domain of u. If kd is choosen to be 2π rather than π, the upper right hand plot in figure 2-3 is obtained. The replicated lobes have fallen into the visible domain of u, resulting in spatial aliasing (the lobes that have fallen into the visible domain are called grating lobes). Thus, the resulting spatial response of a non-steered array exhibits undesireable extra beams (see lower left and lower right plots in Figure 2-3).

To avoid aliasing at endfire [1], the interelement spacing on the microphone array must be $d \leq \frac{\lambda_{min}}{2}$, where λ_{min} represents the wavelength of the highest frequency to be considered in our microphone array [6].

The reason the interelement spacing d is selected to be $\frac{\lambda}{2}$ rather than just λ is because the case of aliasing for a steered beam must be considered. Suppose the beam shown in figure 2-3 is steered to $\theta = \frac{\pi}{4}$. That means that the visible and replicated lobes are shifted. This shift places the replicated lobes closer to the visible domain of u, which in turn means that one need not go as high in frequency as in the non-steered case to bring in a grating lobe.

2.3.2 Beamwidth Variations

The transformation function described in 2.9 not only shows how spatial aliasing occurs, but it also illustrates that the beamwidth of an arbitrary spatial response will vary as a function of both frequency and scan angle [4, 6]. Several definitions of beamwidth are used;

[1] Endfire is defined as steering the main beam to an angle of $\theta = 90$degrees

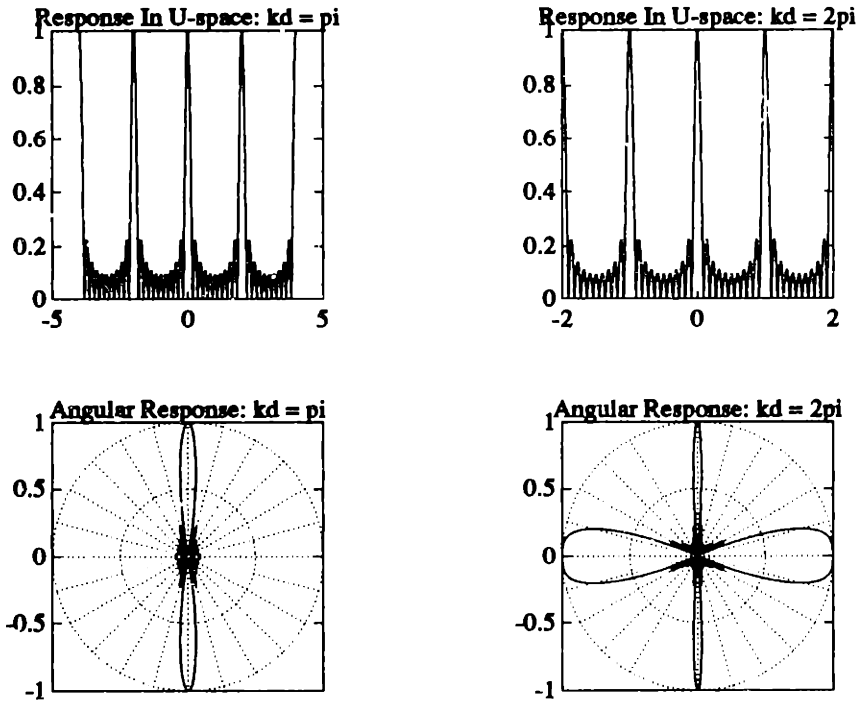

Figure 2-3: Effects of Spatial Aliasing

we will define it as the range in angle θ between the nulls flanking the main lobe. We solve for the nulls rather than the $3dB$ points because the resulting solutions are easier to arrive at.

Before tackling the effects on beamwidth when using the transformation in equation 2.9, let us solve directly for the beamwidth in equation 2.4 when $\{a_n\} = 1$. By setting equation 2.4 equal to zero and solving for the internull distance, $\triangle u$ is found to be [6]:

$$\triangle u = \sin \triangle\theta \simeq 0.88 \frac{\lambda}{(M-1)d} = 0.88 \frac{2\pi c}{(M-1)d\omega} \tag{2.10}$$

It is clear in equation 2.10 that the beamwidth of a non-steered uniformly weighted array is inversely proportional to frequency. As the frequency increases, the beamwidth in u decreases.

By examining the transformation in equation 2.9, the same general result is found. Solving for the nulls flanking the main lobe in the ω_t domain, one null is at ω_{tA} and another

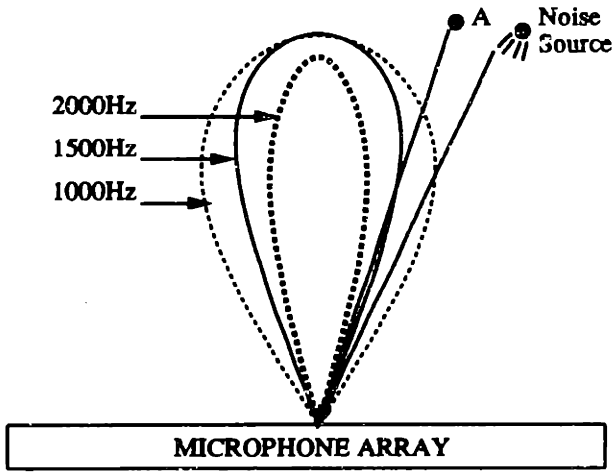

Figure 2-4: Beamwidth Variation as a function of frequency.

at ω_{tB}. Plugging these values for ω_t in equation 2.9, the beamwidth of a similarly weighted microphone array is:

$$\triangle\theta = \theta_A - \theta_B \tag{2.11}$$

$$\theta_A = \sin^{-1}(\frac{\omega_{tA}}{kd}) = \sin^{-1}(\frac{\omega_{tA}c}{\omega d}) \qquad \theta_B = \sin^{-1}(\frac{\omega_{tB}}{kd}) = \sin^{-1}(\frac{\omega_{tB}c}{\omega d})$$

Thus, from equation 2.10 or 2.11, it is clear that the beamwidth varies as a function of frequency.

Beamwidth variations as a function of frequency cause two adverse effects. The first, illustrated in figure 2-4, is that the array picks up substantial off-axis low frequency noise. This effect degrades the performance of the microphone array. The second effect is that signals originating off-axis to the beam are subjected to a lowpass filter. For example in figure 2-4, if source A is broadband, the higher frequencies won't be picked up because the beamwidth has shrunk for those higher frequencies (2000Hz in Figure 2-4).

Beamwidth variations occur not only as a function of frequency, but also as a function of scan angle θ_o [6]. For an equally weighted microphone array, and assuming that the non-steered beamwidth $\triangle u \simeq \frac{\lambda}{(M-1)d}$ then:

$$\triangle\theta \simeq \frac{\lambda}{(M-1)d\cos\theta_o} \tag{2.12}$$

Thus the beamwidth increases as it is steered off broadside (broadside is defined as perpen-

dicular to the microphone array).

2.4 Effects of Temporal Aliasing in Beamforming

Since we are interested in implementing the microphone array digitally (rather than using analog techniques), the effects of temporal aliasing on beamforming must be considered. Temporal aliasing can occur potentially in one of two ways. The first is by undersampling the incoming plane wave. The second is by failing to effectively lowpass filter a signal during interpolation. The latter effect is more pronounced in our implementation. (Chapters 3 and 4 cover the system design and discuss the use of temporal interpolation.)

The following definitions are used in this section [2]: Ω indicates the frequency in Hertz of the spectrum of the continuous time signal; ω indicates the frequency in radians of the incoming plane wave; and ω_s indicates the spectrum of the sampled signal. Also assume that the spectrum of the incoming plane wave is bandlimited at $\Omega = \Omega_n$ (Nyquist frequency) and that the incoming signal is real.

Thus far complex notation has been employed to indicate wave propagation, but no explicit mention of the time varying portion of the plane wave has been made. The complex notation representing a plane wave is derived from the following convention [7]:

$$\cos(\omega t - kdu) = \cos(kdu - \omega t) = Re[e^{j\omega t}e^{-jkdu}] \tag{2.13}$$

where the amplitude of the plane wave is normalized to one. In reality, any broadband sound is a superposition of an infinite number of plane waves with distinct frequencies, but for the sake of clarity we will focus on one plane wave with frequency ω.

Instead of looking at plane wave propagation in terms of kdu, we can look at it in terms of t. By recalling that $k = \frac{\omega}{c}$, the plane wave propagation term can be viewed as the phase of a time dependent signal. Letting $\phi = \frac{du}{c}$, equation 2.13 can be written as:

$$f(t) = \cos(\omega t - \omega\phi) = \cos(\omega(t - \phi)) \tag{2.14}$$

[2] Note that the section discussing the FIR filter analogy used the frequency variable ω_t as a means of illustration, and this should not be confused with the frequency of the impinging plane wave, ω.

where $f(t)$ now represents a plane wave with amplitude equal to one. Rather than looking at $f(t)$ in time, it can be viewed in s-space by obtaining the Laplace transform of $f(t)$ (the Laplace transform operation is described by $\mathcal{L}\{\cdot\}$):

$$\mathcal{L}\{f(t)\} = \mathcal{L}\{\cos(\omega t)\}e^{-s\phi} \tag{2.15}$$

Letting $C(s) = \mathcal{L}\{\cos(\omega t)\}$ and $s = j\Omega$, the above equation can be rewritten to:

$$\mathcal{L}\{f(t)\} = C(j\Omega)e^{-j\phi\Omega} = F(j\Omega) \tag{2.16}$$

where the phase of $F(j\Omega)$ is odd (anti-symmetric about the origin) and the magnitude of $F(j\Omega)$ is even (symmetric about the origin) [9].

If the signal $f(t)$ is sampled at a rate T, we can express the Fourier transform of the sampled signal $x[n] = \cos(\omega(nT - \phi))$ as [8]:

$$X(e^{j\omega_s}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_c(j\frac{\omega_s}{T} - j\frac{2\pi k}{T}) \tag{2.17}$$

If the incoming signal $f(t)$ is undersampled such that $T > \frac{\pi}{\Omega_n}$, aliasing will occur. By undersampling, replicated segments of the spectrum are introduced into the region $-\pi < \omega_s < \pi$. Looking at the region $0 < \omega_s < \pi$ and recalling the original spectrum $F(j\Omega)$ is bandlimited, aliasing in this region can be described as:

$$X(e^{j\omega_s}) = C(j\frac{\omega_s}{T})e^{\frac{-j\phi\omega_s}{T}} + C(j\frac{\omega_s}{T} - j\frac{2\pi}{T})e^{-j\phi(\frac{\omega_s}{T} - \frac{2\pi}{T})} \tag{2.18}$$

where the above equation is valid for the region $T(\Omega_s - \Omega_n) < \omega_s < T\Omega_n$ and $\Omega_s = \frac{2\pi}{T}$. Since the phase of the incoming signal is odd the second term in equation 2.18 can be written as:

$$C(j\frac{\omega_s}{T} - j\frac{2\pi}{T})e^{j\phi(\frac{\omega_s}{T} + \frac{2\pi}{T})}$$

Flipping back into the continuous time spectrum Ω, and plugging the above result into

equation 2.18, the aliased signal $\hat{F}(j\Omega)$ is given by:

$$\hat{F}(j\Omega) = C(j\Omega)e^{-j\phi\Omega} + C(j\Omega - j\Omega_s)e^{j\phi(\Omega+\Omega_s)} \tag{2.19}$$

If we obtain the inverse Laplace transform of equation 2.19 and look at the region $\Omega > 0$ we find that:

$$\hat{f}(t) = \cos(\omega(t-\phi)) + \alpha(\Omega)\cos(\omega(t+\phi)) \tag{2.20}$$

where $\alpha(\Omega) = 0$ for $\Omega < \Omega_s - \Omega_n$ and $\alpha(\Omega) = m(\Omega)$ for $\Omega_s - \Omega_n < \Omega < \Omega_n$, and $m(\Omega)$ is some function of Ω which quantifies the amount of aliasing. Depending on the frequency ω of the impinging wave, more or less aliasing will occur. For example, if the frequency of the impinging wave is not in the region defined by $\Omega_s - \Omega_n < \Omega < \Omega_n$, then there is no aliasing. If the frequency is in the region defined by $\Omega_s - \Omega_n < \Omega < \Omega_n$, then there will be an effect. With this result, and using equations 2.7 and 2.13, we can write the general form for the spatial response of a steered beam with temporal aliasing:

$$H(k,u) = \sum_{n=-N}^{N} a_n e^{-jkd(u-\beta)} + \gamma(\Omega) \sum_{n=-N}^{N} a_n e^{jkd(u-\beta)} \tag{2.21}$$

$$\beta = \sin\theta_o$$

where $\gamma(\Omega)$ describes the amount of aliasing. For the case of undersampling, $\gamma(\Omega) = \alpha(\Omega)$ from equation 2.20. The derivation shown in this section solves for the case of aliasing as a result of temporal undersampling (equation 2.20). The final result shown in equation 2.21 is generalized to include all types of aliasing (not only from undersampling, but from interpolation and downsampling) by altering the scaling function $\gamma(\Omega)$. Furthermore, the above derivation can be extended to a superposition of plane waves constituting a sound wave.

Equation 2.21 indicates that if the time signal is aliased a beam is introduced at a symmetric angle to the intended steering direction. This effect is of dire consequence when implementing a steerable microphone array since imperfect filtering during the interpolation stage will result in shoddy beamforming at high frequencies. Ideally, the value of $\gamma(\Omega)$ should be kept as close to zero as possible for all frequencies.

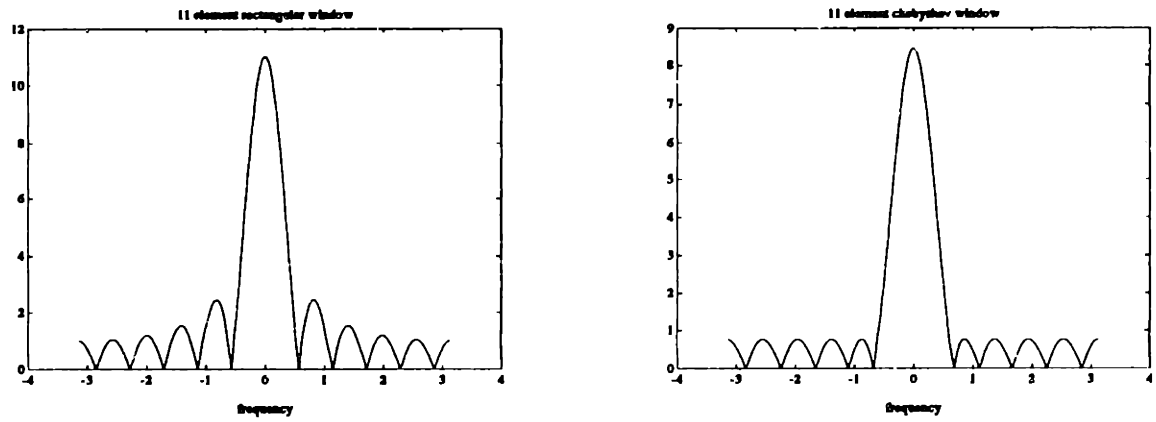

Figure 2-5: Rectangular Window vs. Chebyshev Window

2.5 Array Shading Coefficients

The elemental weights a_n that were introduced at the beginning of this chapter are usually referred to as shading coefficients. These coefficients determine the spatial response of the array. We would like to select a set of a_n in our array implementation such that a satisfactory trade-off occurs between beamwidth and out of band attenuation. This problem can be addressed directly from results in FIR lowpass filter design [8].

It has been shown [6, 10] that the optimal tradeoff between beamwidth and sidelobe level is attained using a set of shading coefficients which match those determined by a Chebyshev window. An equiripple minimization algorithm is used, resulting in sidelobe attenuation that is constant out of band. Figure 2-5 shows the Fourier transform in ω_t space for a rectangular window (left plot) and a Chebyshev window (right plot).

Either by applying the transformation in equation 2.9 to the result of solving for the nulls in equation 2.8 or by solving directly for the nulls flanking the main lobe in equation 2.6, the beamwidth for rectangular window shading coefficients is [4, 12]:

$$\Delta\theta = 2\sin^{-1}\left(\frac{2\pi}{M\omega\tau_o}\right) \tag{2.22}$$

where $\tau_o = \frac{d}{c}$ and $M = 2N + 1$ is the number of elements in the array. If the shading

coefficients are set to match a Chebyshev window, the beamwidth is:

$$\triangle\theta = 2\sin^{-1}\{\frac{2}{\omega\tau_o}\cos^{-1}(\frac{1}{x_o}\cos(\frac{\pi}{4N}))\} \tag{2.23}$$

where $x_o = \frac{1}{M}\cosh^{-1}(R)$ and R is the ratio of main beam amplitude to the sidelobe amplitude. Thus, in an attempt to achieve good sidelobe attenuation without compromising beamwidth, the Chebyshev shading coefficients were selected for our microphone array.

Chapter 3

Beamformer Design

3.1 System Architecture

Figure 3-1 shows the system architecture of the digital beamformer. The system should be designed to: receive an acoustic signal (microphone array); perform all necessary preamplification, anti-alias filtering, and sampling (hardware); beamform (software); and provide the output to a loudspeaker device connected to the system. All hardware and low-level software design should be transparent to the user. Furthermore, the user should be able to control the beamformer settings (i.e. automatic vs. manual steering).

3.2 Hardware Architecture

3.2.1 Microphone Array

In chapter 2 we discussed the frequency dependence of beamwidth in microphone arrays. Equations 2.10 and 2.11 illustrate that the greater the bandwidth of our array (or the shorter the interelement distance d), the greater the variation in the beamwidth. If d is set to avoid aliasing at 8000Hz, a narrow beam will result at 8000Hz, but a very wide beam will result at 1000Hz. In an attempt to avoid such beamwidth differences, the use of nested arrays is suggested [2].

Figure 3-2 shows the design of a nested array. Nested arrays minimize beamwidth variation over a wide frequency band by breaking up the wide band into smaller bands.

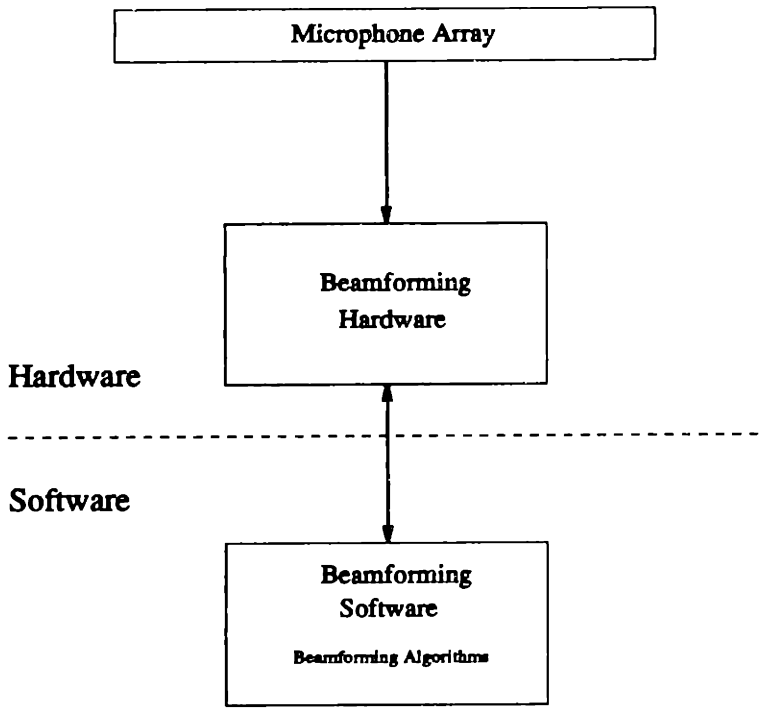

Figure 3-1: System Architecture

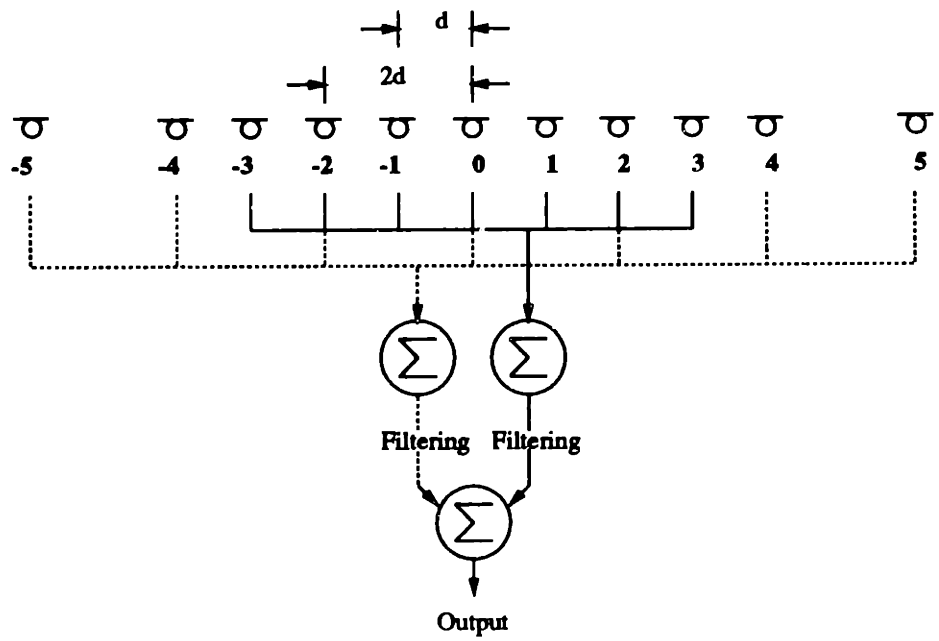

Figure 3-2: Nested Array

In *harmonic nesting*, a set of microphone arrays are superimposed, each satisfying the requirement $d \leq \frac{\lambda_{min}}{2}$ for the frequency range of interest. All frequencies not in the range of the microphone subarray are filtered out. In figure 3-2, elements -3, -2, -1, 0, 1, 2, 3 form one array, while elements -5, -4, -2, 0, 2, 4, 5 form a second array (note that elements -2, 0, 2 overlap). The elemental outputs for each subarray are appropriately added, filtered, and recombined.

An important consideration in *harmonic nesting* is to ensure that the origin of all nested subarrays is common. Without a common origin *harmonic nesting* will lead to phase centering problems which in turn leads to comb filter effects on the received signal.

Suppose that the origin of the longer array in figure 3-2 is set to element 5, while the origin of the shorter array is set to element 3. The resulting response of the combination of both arrays will be:

$$H_{nested\,array} = H_{array\,1} + H_{array\,2} = \tag{3.1}$$

$$\sum_{n=0}^{M} a_n e^{-jknd_{array1}\sin\theta} + e^{-jk(d_{array1}+d_{array2})\sin\theta} \sum_{n=0}^{M} a_n e^{-jknd_{array2}\sin\theta}$$

where d_{array1} is the interelement spacing of the long array, and d_{array2} is the interelement spacing of the short array. Equation 3.1 exhibits a comb filter effect because of the delay $e^{-jk(d_{array1}+d_{array2})\sin\theta}$.

In our proposed beamformer, a twenty-nine element nested microphone array is used. The array is the combination of four nested arrays. Each of the nested arrays (subarrays) is comprised of eleven elements. The spacing of the elements was designed to divide the 250-8000Hz frequency band into four octaves. Recalling from chapter 2 that the required spacing to avoid spatial aliasing was $d \leq \frac{\lambda_{min}}{2}$, the following spacings were selected for each of the nested arrays: .02m for the 4000-8000Hz octave, .04m for the 2000-4000Hz octave, .08m for the 1000-2000Hz octave, and .16m for the 250-1000Hz octave. To illustrate the calculations performed to obtain the interelement spacing, note that for the high frequency octave $\lambda_{min} = \frac{c}{8000}$, thus $d = \frac{c}{16000} \simeq .02m$.

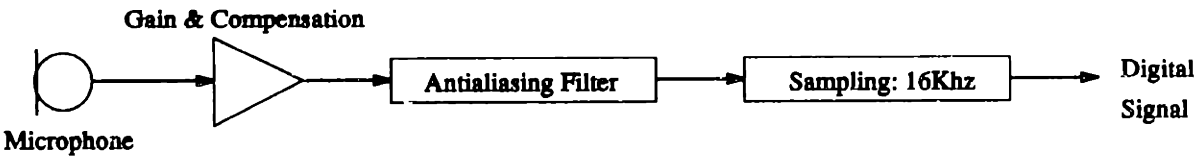

Figure 3-3: Signal hardware path.

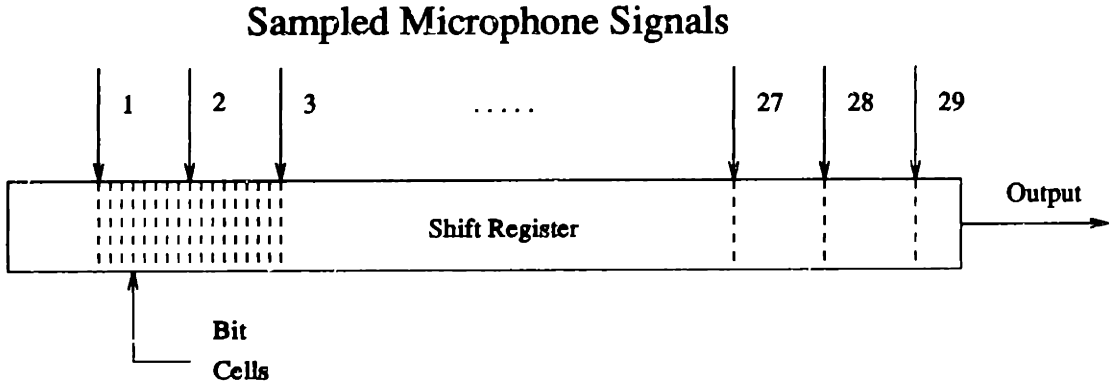

Figure 3-4: Parallel to serial interface.

3.2.2 Microphone Signal Preamplification and Sampling

The design of the preamplification hardware is relatively straightforward. An analog gain and compensation stage are needed to process the low-level microphone response. Compensation will be used to flatten out the microphone frequency response (which has a high pass filter characteristic [11]) and provide a signal level of one volt for a 100 dB SPL (sound pressure level) at the microphones.

The sampling hardware follows compensation and includes an anti-alias filter and a codec (code/decoder chip) (see figure 3-3). Anti-alias filtering is necessary to avoid temporal aliasing as discussed in the previous chapter. The codecs operate at a 16kHz sampling rate to achieve a bandwidth of 8kHz.

The last stage in the hardware design is to multiplex all the microphone signals onto a single path. This is necessary because we are using the serial port on AT&T DSP3210 DSP chips [5] as the input stage. The resulting design to effectively multiplex all twenty-nine sampled signals is to feed each signal into a portion of a shift register (as shown in figure 3-4). The sampled bits will then be available in serial fashion at the output of the shift register.

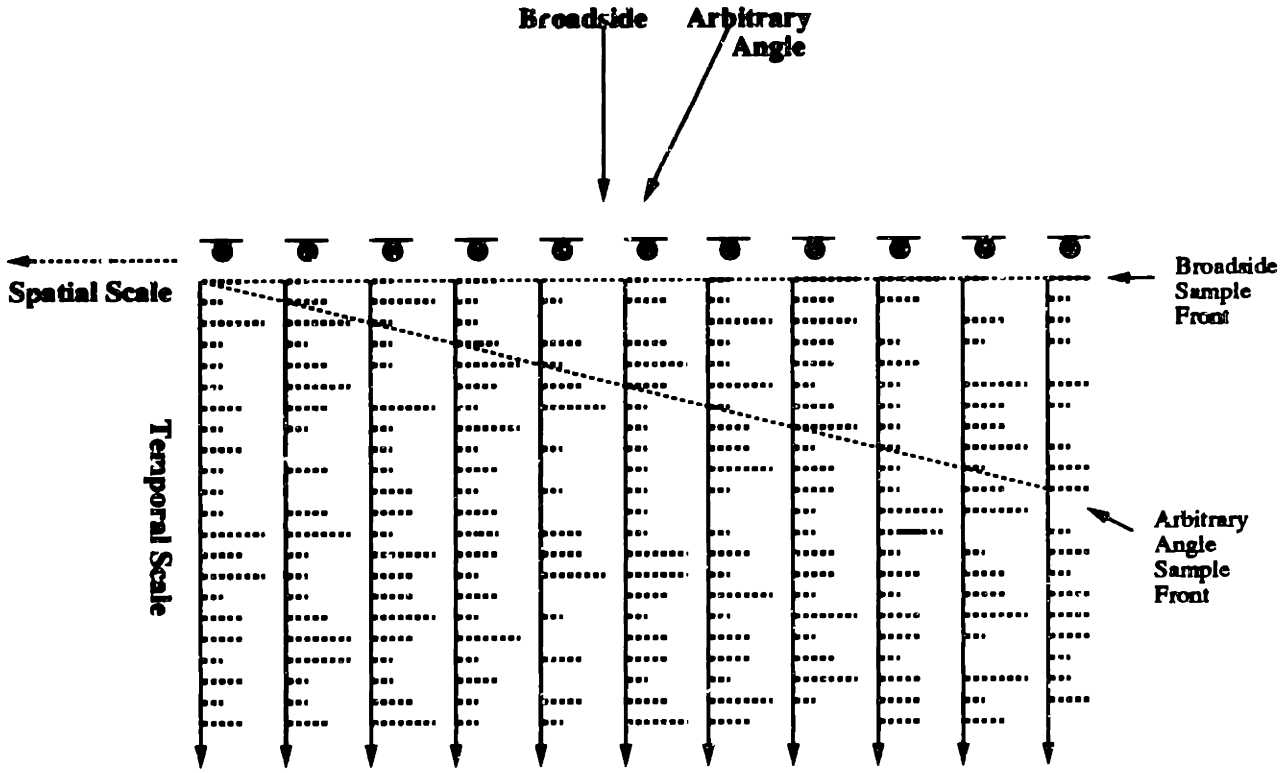

Figure 3-5: Beamforming Visualization

3.3 Software Architecture

3.3.1 Beamforming

A good way of understanding how to implement beamforming in our system is to visualize the microphone samples as the continuation of a planar wavefront. Figure 3-5 schematically pictures the continuation of the wavefront in reference to the microphone samples. To form a beam at broadside, all of the current samples would be added (illustrated with the horizontal line in figure 3-5). To beamform at an arbitrary angle, all of the samples that correspond to that wavefront would be added (note that the wavefront of the arbitrary angle includes the appropriately delayed microphone samples).

Several issues arise when observing figure 3-5. First, which has been discussed briefly in the preceding section, is the need to sample the incoming wave at twice the Nyquist frequency. Second, from figure 3-5, sufficient memory must be provided to hold all the necessary previous samples that may be needed to perform beamforming. Third, when the

beamformer is steered to an arbitrary angle, the delay between microphone samples must be known precisely to pick the correct sample in memory. Since the sample rate is not necessarily equal to the intermicrophone delay, the normalized delay between microphones must be determined. For example, if samples were taken every second and the intermicrophone delay were 1.5 seconds, then the normalized delay would be 1.5. Specifically, the normalized delay can be defined as:

$$\tau_{norm} = \frac{d \sin \theta}{c T_{samp}} \tag{3.2}$$

Notice that τ_{norm} is not necessarily an integer, in which case interpolation is required to obtain the desired sample. By appropriate selection of the steering angle θ, the number of samples that must be interpolated can be minimized.

After beamforming using the above method, the undesired frequencies for each subarray must be filtered out. Bandpass filtering will be implemented using IIR filters; the required interpolation will be implemented using FIR filters. These choices provide good trade-offs between precision and real-time implementation.

Due to real-time constraints, the most efficient way to perform beam steering (both manual and automatic) is to simultaneously implement N beams, where each beam corresponds to a given steering direction. In our case, we have chosen seven beams steered at the angles -54, -32, -16, 0, 16, 32, and 54 degrees. As will be shown in Chapter 4, these angles minimize but do not eliminate the need for interpolation.

3.3.2 Beam Steering: Manual and Automatic

Forming seven simultaneous beams greatly simplifies the problem for manual steering. The desired beam direction is accomplished by selecting one beam On and turning all other beams Off. For automatic steering, having formed seven beams provides a good platform on which to run a "voting" algorithm to select the desired beam.

To select the appropriate beams, a simple energy detection scheme is employed. This procedure is simple to implement and does not sacrifice performance. A complex voice detection scheme used in a prior digital implementation was not very promising [3].

nested arrays are denoted LF (Low Frequency) for 250-1000Hz, MLF (Mid-Low Frequency) for 1000-2000Hz, MHF (Mid-High Frequency) for 2000-4000Hz, and HF (High Frequency) for 4000-8000Hz.

3.3.3 Noise

The beamformer design should try to account for ambient noise. A method that helps minimize the effects of ambient noise (degraded acoustic performace) is to estimate background noise when the system starts up (preferably with people in the room who remain quiet for the duration of the background noise estimate). Once a background noise estimate has been performed, it can be used as a threshold to reduce the probability of causing the automatic beamsteering algorithm to switch on noise. A simple estimate of the background noise is to run an n second energy average in the teleconferencing room. The resulting value can then be passed on to the automatic beamsteering algorithm and used as a threshold prior to the "findmax routine" indicated on the lower right hand corner of figure 3-5.

3.3.4 Microphone Mismatch

In designing the software to run on the DSP chips and the PC, it is desirable to minimize the impact of component mismatch. It is well known that the microphones cannot be expected to exhibit identical responses. In an attempt to help reduce the microphone mismatch, normalization of the twenty-nine microphones to a predetermined reference is suggested.

This is accomplished by presenting an acoustic tone and obtaining an n second energy average of the output of each microphone in the array. This provides satisfactory performance in an anechoic chamber; a broadband source would be required in a reverberant environment. If all the resulting values are normalized to the average energy measured in the center microphone, mismatch can be approximately cancelled out with the determined scale factor. For example, running this algorithm on the implemented beamformer yielded a set of normalization constants which were within 10% (1dB) of each other.

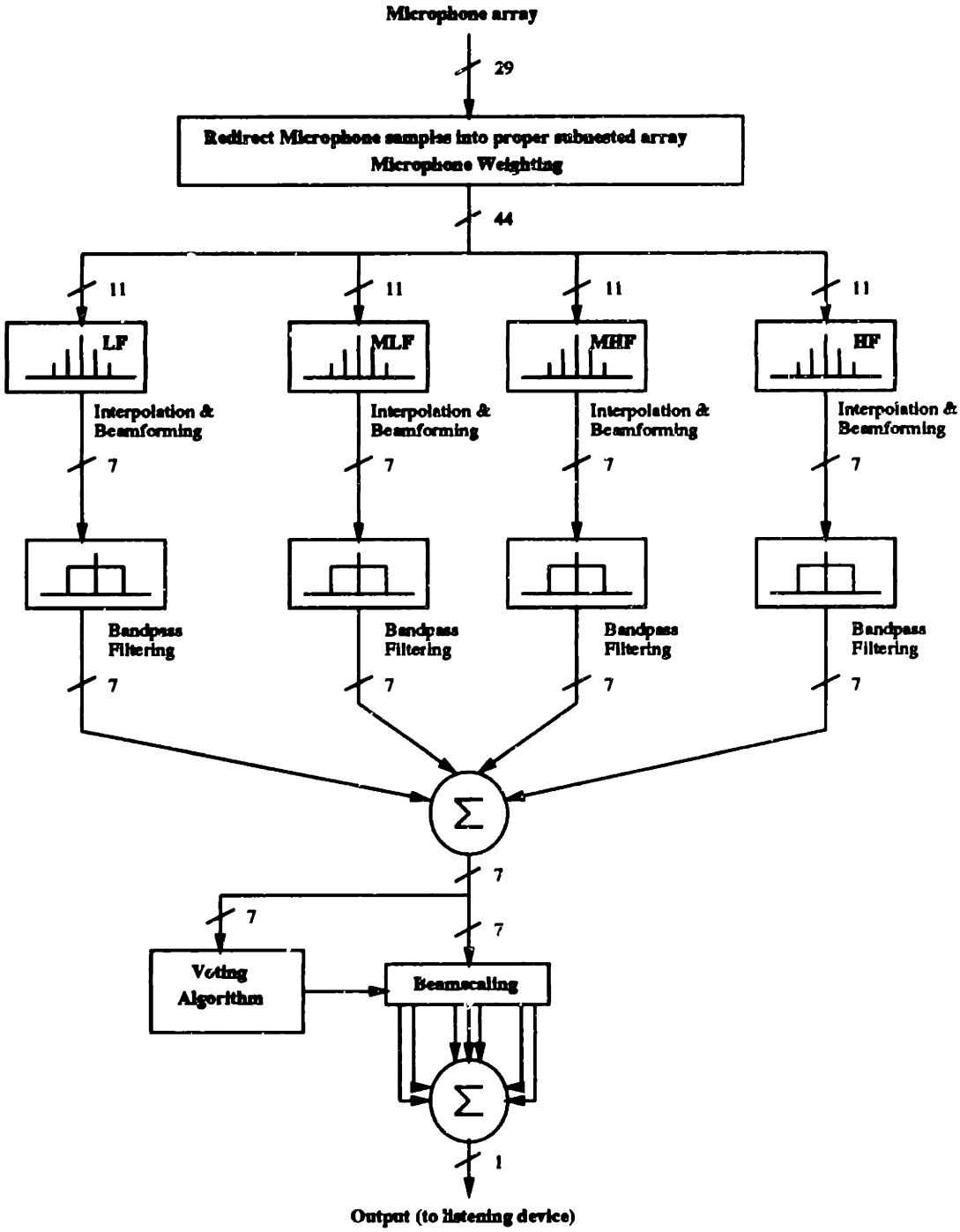

Figure 3-7: Beamforming and Voting algorithms

Chapter 4

Digital Beamformer Implementation

This chapter describes the implementation of the digital beamformer. The preamplification stage will be discussed only briefly since the gain and compensator circuits are not novel in their implementation. Most of the chapter will be spent describing the beamforming algorithm and the DSP-sampling hardware interface. Some of the difficulties encountered in using the MP3210 DSP boards designed and fabricated by Ariel Corporation [13] will also be discussed.

4.1 Overall System Description

Figure 4-1 illustrates the system configuration and information flow. Note that there are three distinct sections (also refer to figure 3-1 in chapter 3): hardware, firmware, and software.

The hardware section is comprised of the microphone array, preamplification hardware, anti-alias filtering hardware, and sampling hardware. The hardware section also includes the speaker which serves as the system output device. The preamplification hardware is housed apart from the anti-alias filtering and sampling hardware.

The firmware section has been made separate from the hardware section to explicitly distinguish the function of the AT&T DSP3210 chips. Since the coding of the DSP chips in

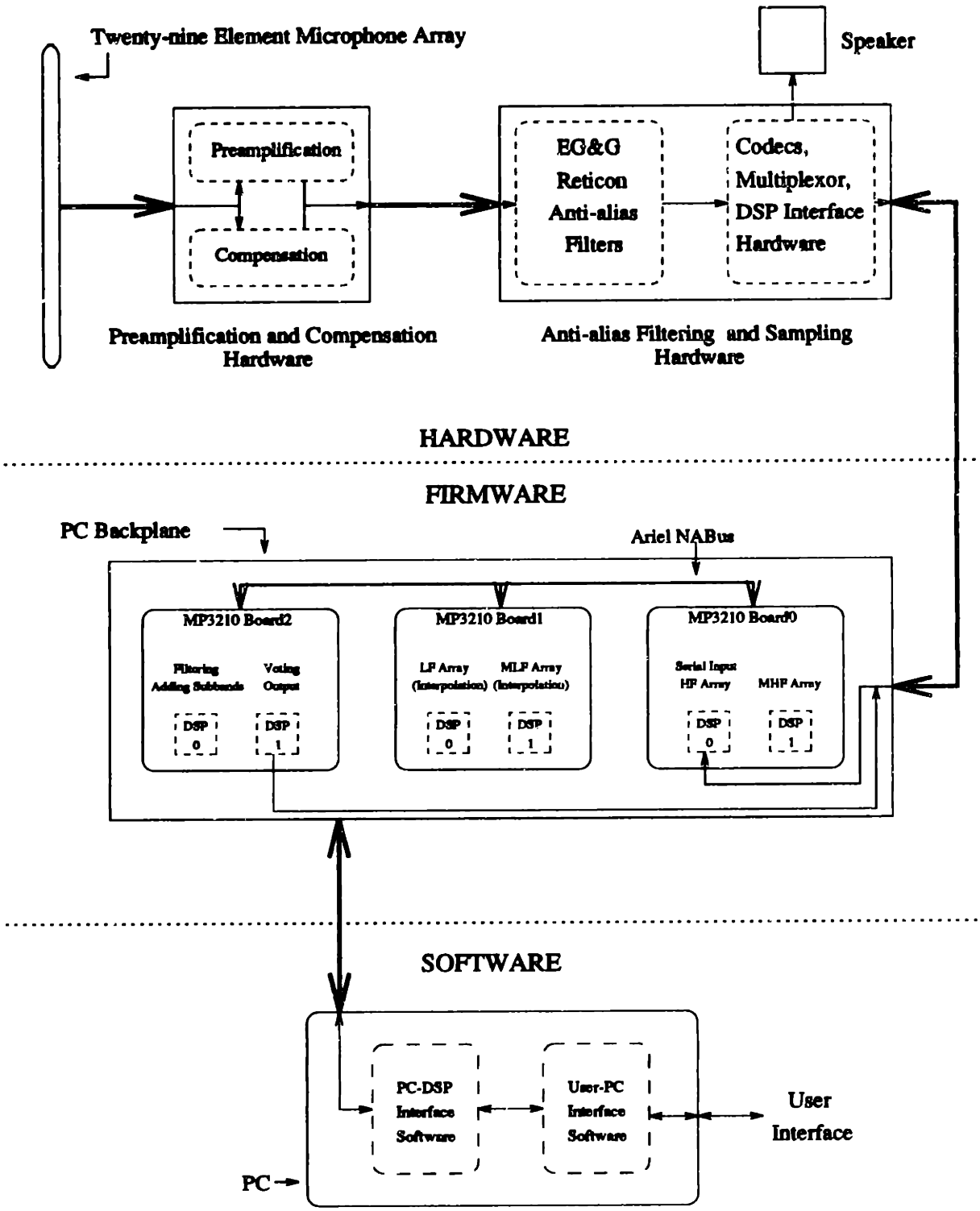

Figure 4-1: Overall System Implementation

assembly language requires explicit knowledge of the surrounding hardware, it is appropriate to call this section firmware since it involves both hardware and software. The firmware section sits inside the PC in the form of plug-in boards to the PC backplane.

The third section in figure 4-1 describes the software aspect of the system. This refers to the programs sitting in the PC memory which are responsible for the user interface and the PC-MP3210 board interface.

The overall signal flow in the system is as follows: Sound is received at the microphone array. The microphone output signals are sent to the preamplification hardware. The resulting amplified signals are then anti-alias filtered, sampled, and multiplexed onto a single channel. Timing hardware, housed together with the sampling hardware, controls the transmission of the multiplexed microphone signals. The microphone samples are received in DSP0 on MP3210 Board0. This DSP chip is responsible for splitting the incoming signal into the four subbands as indicated in figure 3-7. The subbands are then processed on four distinct DSP chips: DSP0 on Board0 processes the HF array, DSP1 on Board0 processes the MHF array, DSP0 on Board1 processes the LF array, and DSP1 on Board1 processes the MLF array. The results of Board0 and Board1 are sent to DSP0 on Board2, where the four subarrays are filtered (for each subband to avoid aliasing) and recombined. The seven resulting formed beams are sent to DSP1 on Board2 which determines the steering direction and returns the final beamsample which is reconstructed and routed to the speaker.

The software stage is responsible for ensuring that the assembly code is running properly on the MP3210 boards and provides a user interface to control certain beamforming parameters.

4.2 Microphone Array and Preamplifier Implementation

The microphone array was built using twenty-nine first order differential microphones [16] laid on a metal bar following *harmonic nesting* principles described in chapter 3. There are four subarrays in the microphone array, each consisting of eleven microphones. Due to the subarray overlap, only twenty-nine microphones are required. The total span of the microphone array is approximately 1.6 meters, corresponding to ten times the intermicrophone distance of the lowest octave array (see chapter 3 for a description of the array spacing).

The microphones used are noise cancelling electret microphones with a $\cos\theta$ reception pattern (where θ is defined in chapter 2). Differential microphones were used for two reasons: historically, since the microphone array had been previously built, and for the advantage that they provide a higher directivity for sound originating at angles up to about 60 degrees. The microphone outputs are low-level signals that require amplification and compensation. Compensation and gain are performed in separate stages. Both the gain and compensator circuits use the 5532 internally compensated dual low noise operational amplifier. The compensation attempts to flatten out the microphone response which has a high pass filter characteristic. The gain stage provides one volt output for a 100db SPL (sound pressure level) at the microphones.

The $\cos\theta$ radiation pattern of the differential microphones has one negative consequence for our array. The attenuation associated with the $\cos\theta$ radiation pattern requires us that off-broadside steered beams be scaled by a factor $\frac{1}{\cos\theta}$.

4.3 Sampling Hardware/DSP Interface

After amplifying and compensating the microphone outputs the signals are sampled at 16kHz for an 8kHz bandwidth. To avoid aliasing, lowpass filtering is performed prior to sampling. A switched-capacitor anti-aliasing filter from EG&G Reticon (RF6609A) was used. The analog filter outputs are then routed to PCM codecs from AT&T Microelectronics (T7525), after which there are twenty-nine 16 bit digital signals. These digital signals are multiplexed and sent to DSP0 on Board0 using a custom designed high speed multiplexer [17].

To successfully accomplish the above scheme, careful timing signals must be provided to the T7525 codecs, the high speed multiplexer, and the DSP3210 chip located in the PC. The timing signals are generated on a PAL chip housed together with the sampling hardware. These timing signals are sent to the Codecs and the high speed multiplexer through board traces. DSP timing signals originating in the PAL are sent to DSP0 on Board0 and DSP1 on Board2 through a ribbon cable connecting the sampling hardware to the MP3210 board. This ribbon cable also transmits the multiplexed microphone signals. Figure 4-2 provides a schematic of this setup.

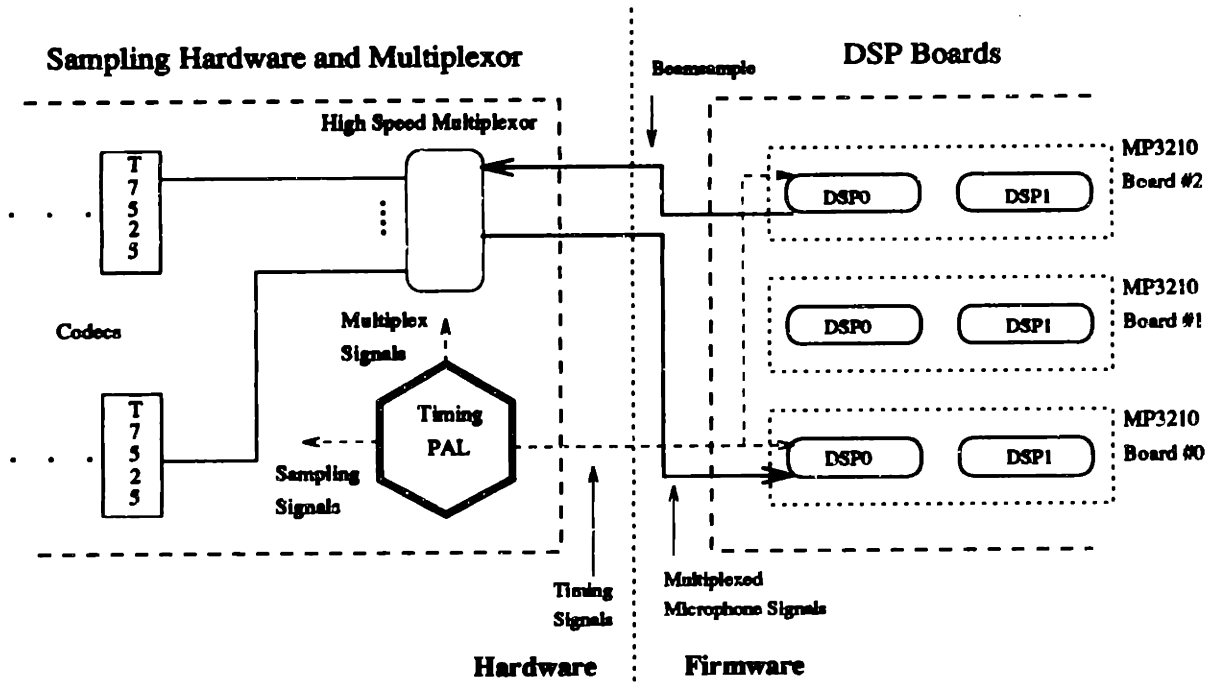

Figure 4-2: Interface Schematic

The configuration of DSP chips inside the PC is also noteworthy. The system runs in a 486 PC holding three Ariel MP3210 boards. Each Ariel board contains two DSP3210 chips and external SRAM and DRAM. There are a total of six DSP3210 chips inside the PC. Figures 4-1 and 4-2 illustrate DSP0 on Board0 receiving the serial data from the high speed multiplexer while DSP2 on Board2 sends the final beamforming result back to a codec. The DSP3210 chips require six signals to interface with the incoming serial data [5]. The timing signals and data are sent over a ribbon cable. Due to the high frequency clock employed to interface with the DSP chip (16MHz), it was necessary to construct a daughter card to properly terminate the ribbon cable connection and clear up the timing signals for the DSP chip. Without the daughter card, the timing signals arriving at the DSP3210 chip were useless. The daughter card was comprised of a resistor pack used to terminate the ribbon cable and an inverting buffer used to drive the timing signals to the DSP chips.

4.4 Beamforming Algorithm Implementation

4.4.1 MP3210 Board Design

The MP3210 board [13] is equipped with two DSP3210 chips (as mentioned above). The DSP3210 chips have a 50MHz clock rate and a limited amount of internal SRAM (2Kbytes). The MP3210 boards provide off-chip DRAM and SRAM. Unfortunately, all off-chip memory is shared between both processors, partially defeating the purpose of placing two processors per board. Due to the shared external memory, the DSP chips are not free to run in parallel unless all assembly code routines are cached into the internal DSP SRAM making all external SRAM and DRAM superfluous.

The MP3210 boards were designed to be connected together through the New Ariel Bus (NABus), a PC independent bus providing board-to-board communications when using more than one board. In an attempt to make the NABus as general as possible, the designers unfortunately overlooked the cost to DSP efficiency by causing all board-to-board memory reads to interfere with I/O DSP3210 processes. This system design severely limited the functionality of the NABus in our application. Beyond the poor NABus-DSP interface, the designers also failed to account for system clock phase delays caused by propagating one clock signal to all the MP3210 boards used.

Finally, the MP3210 board designers failed to provide input buffers for all external signals. This failure led us to build separate daughter cards to interface to the MP3210 boards. It should also be noted that beyond basic design flaws, the MP3210 boards were unreliable. It was found that the main limitation in our implementation was the shoddy quality and design of the MP3210 boards as evidenced by repeated failures. Common failures included non-deterministic NABus behaviour, poor chip socketing, and failure to provide buffer circuitry to avoid DSP chip latch-up with power outages.

4.4.2 Implementation Architecture

Three boards were used, each with two DSP3210 chips. The boards were connected to each other through the NABus. The NABus provided for all data transfer between boards. The resulting architecture split the beamforming algorithm into modules, each

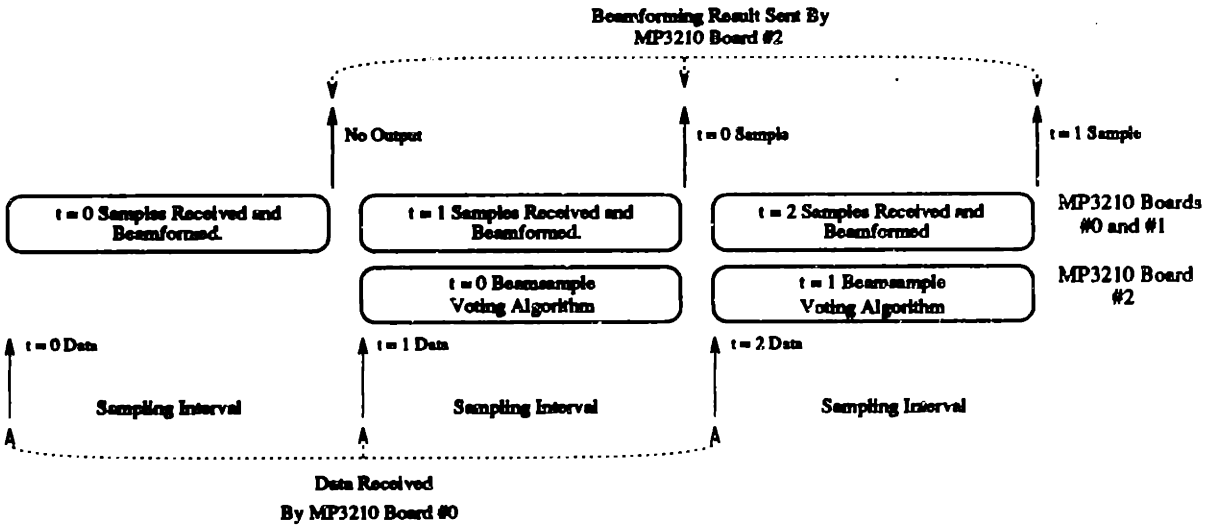

Figure 4-3: Implementation Architecture

module running on one MP3210 board.

A sampling rate of 16kHz requires that an output be provided every 62.5 μs. This time period does not allow sufficient time to perform all the necessary computations required in the beamformer; therefore, a pipelined architecture was introduced (see figure 4-3). Beamforming takes place in one sampling interval, and voting takes place in the next sampling interval. This format allows for a total of 125 μs for each sample.

The assembly code was split up into the following modules:

1. Data acquisition, data redirection, and HF (High Frequency) array processing.
2. MHF (Mid-High Frequency) array processing.
3. MLF (Mid-Low Frequency) array processing.
4. LF (Low Frequency) array processing.
5. Subarray filtering and recombination.
6. Voting algorithm and beamforming output.

The first and second modules ran on MP3210 Board0. The third and fourth modules ran on MP3210 Board1. The fifth and sixth modules ran on MP3210 Board2. The final beam

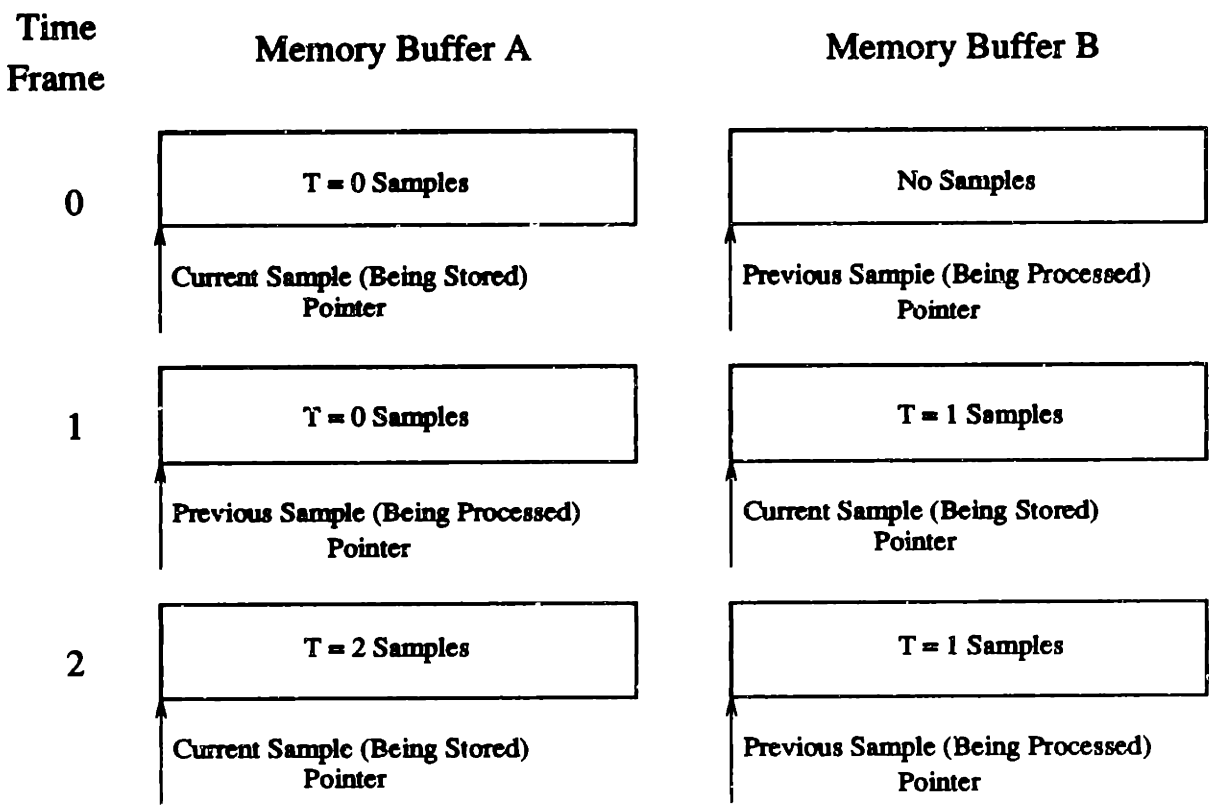

Figure 4-4: Memory Handling: Data Acquisition

sample is available from MP3210 Board2. Figure 4-3 illustrates the pipelined architecture of the algorithm implementation. Note that it is MP3210 Board2 that is pipelined.

All assembly code written was cached into the DSP chips at startup avoiding conflict states arising from the shared external memory on the MP3210. All NABus memory reads were kept as short as possible to minimize the nondeterministic effect this had on the I/O DSP3210 processes.

4.4.3 Memory Handling

There were two areas that required special considerations for memory handling. The first was to provide for memory storage during data acquisition. The second was to provide for memory storage to hold all necessary microphone samples to reconstruct the determined wavefronts.

The data acquisition memory storage is implemented by exploiting the direct memory access (DMA) function of the DSP3210 chip. The chip in charge of data acquisition stores

the incoming microphone samples as a background process, while it redirects the microphone samples which arrived in the previous time frame. This means that samples are not processed in the same time frame in which they arrive. Memory storage is handled by storing the currently received samples in one buffer, while the previously received samples are read from a second buffer. At the end of the time frame, the memory pointers switch buffers. Figure 4-4 shows how the memory pointers are updated for the input data memory storage.

The memory storage for holding microphone samples for a given subarray is just big enough to ensure that we have the first and last sample needed to reconstruct a wavefront. We then simulate linear memory by performing a simple memory pointer updating procedure which resets the pointer to the base of memory when a flag is set.

4.4.4 Beamforming Algorithm: Interpolation and Bandpass Filtering

Chapter 3 discussed the need for interpolation and bandpass filtering in the beamforming algorithm. Interpolation is necessary to reconstruct wavefront samples that do not match the actual samples. Bandpass filtering is necessary to remove undesired frequencies for each of the four subarrays. This filtering helps to both avoid spatial aliasing (by filtering out the high frequencies) and wide low frequency beamwidth (by filtering out the low frequencies).

Chapter 3 introduced the fact that the microphone samples received will not always correspond to the actual intermicrophone delay. Therefore a set of angles must be determined that allow for integral intermicrophone delays. If the intermicrophone delay is non-integral, interpolation will be necessary to determine the wavefront sample of interest. The angles requiring the least amount of interpolation can be determined with equation 3.2. The approach used to obtain this set of angles was to determine angles for the LF octave such that the intermicrophone delay equaled an integer (thus requiring no interpolation). Furthermore, the angles had to be spaced at about twenty degree increments of each other for proper coverage.

By using the above constraints and keeping in mind that only seven beams must be formed, it was found that $\frac{cT_{samp}}{d_{LF}}$ equalled an integral amount for the angles $\theta = 16, 32, 54$. Having thus determined the angles based on the LF octave, the intermicrophone delays

Θ	HF	MHF	MLF	LF
0°	0	0	0	0
16°	.25	.5	1	2
32°	.5	1	2	4
54°	.75	1.5	3	6

Figure 4-5: Normalized Intermicrophone delays.

for the remaining three octaves can be calculated with the use of equation 3.2. Figure 4-5 shows the calculated normalized intermicrophone delays for the LF,MLF,MHF, and HF octaves. Note that interpolation will be needed when determining the wavefront samples for the steered beams in the HF and MHF octave. The LF and MLF octave have integral delays thus no interpolation is required there.

Interpolation on the HF and MHF array is accomplished by designing a set of lowpass FIR filters with cutoffs determined by the non-integral delay [8]. For the HF octave's 16- and 54-degree steering directions, a lowpass filter with cutoff $\frac{\pi}{4}$ was determined using a Hamming window. For the HF octave's 32-degree steering direction and MHF octave's 16- and 54-degree steering directions, a lowpass filter with with cutoff $\frac{\pi}{2}$ was determined using a Hamming window. The interpolation was then performed using polyphase filtering [18].

In chapter 2, the effects of temporal aliasing while beamforming were discussed. It is appropriate here to mention that FIR interpolators without sharp edges can cause the type of aliasing that was discussed in chapter 2. Due to the time constraints imposed on the system, it was found that the interpolators used were ineffective at the highest frequencies (7200-8000Hz). Therefore, the bandwidth of the system was scaled back to be between 250-7200Hz. The FIR interpolators were causing high frequency aliasing for the steered beams (since interpolation is used only for the steered beams). The effect can be described in terms of equation 2.21 where $\gamma(\Omega) \neq 0$ for $\Omega > 7200$ Hz.

After reconstructing the wavefront, each subarray output is bandpass filtered. Bandpass filtering is achieved using fourth-order edge Cauer (elliptic) filters designed using Matlab

[19]. IIR filters are used for bandpass filtering because sharp edges are required. IIR Cauer filters provide sharper edges with a lower order in comparison to FIR filters.

4.4.5 Voting Algorithm: Parameter Determination

Apart from the actual beamforming implementation, the steerable microphone array required the implementation of a voting algorithm to select a steering direction. As was described in chapter 3, the voting algorithm uses first-order lowpass filters to obtain the energy average and the beamscale coefficients.

The parameter determination for these lowpass first order IIR filters used the impulse invariance design technique [8]. The resulting first order lowpass filter can be described as:

$$H(Z) = \frac{G}{1 - PZ^{-1}} \tag{4.1}$$

where $G = 1 - P$, $P = e^{\frac{-T}{\tau}}$, τ is the time constant of the lowpass filter in seconds, and T is the sampling period of the lowpass filter in seconds.

With this knowledge, the lowpass filter coefficients can be determined. It was empirically determined that 10 msecs is an adequate period to average energy. The *Turn On* filter parameters were determined by assuring that whenever the beamformer switches beams, the new direction is to be audible quickly (fast rise time). A 1 msec time constant provides a good first estimate. For the *Turn Off* filter parameters, the opposite is true and thus, a slow fall time is desired. A 1 sec time constant provides a good first estimate.

Two other important parameters are necessary in the voting algorithm, a noise estimate and the hysteresis offset value to avoid random beam switching (see chapter 3). The background noise estimate is obtained by running a program which averages the received signal in the room for approximately 2 seconds prior to beamforming. This estimate is then incorporated into the voting algorithm. The hysteresis offset value was implemented as a user controllable parameter. The offset value is determined empirically in the current implementation and tends to be approximately equal to half the maximum determined energy (that means that the potential new steering direction must have an energy approximately 3dB bigger than the current steering direction's energy). The offset value is chosen empirically

because the time constraints on the voting algorithm did not allow for extra computation time.

All voting algorithm parameters can be controlled and altered by the user through the user interface. Thus, the voting algorithm can by fine tuned for any desired application.

4.4.6 DSP Chip Utilization

The amount of signal processing needed to implement the beamformer required the use of six DSP chips. A good benchmark of how efficiently the DSP chips are being used is the percentage of instruction cycles used out of the total number of instruction cycles available for computation.

Board0 and Board1 are used close to full capacity, while Board2 is used less optimally. DSP0 and DSP1 on Board0 are used to about 85% capacity. DSP0 and DSP1 on Board1 are used to about 85% capacity as well. DSP0 on Board2 is used to about 45% capacity while DSP1 on Board2 is used to about 85% capacity. It was determined that this break-up in the beamforming algorithm provided the most efficient use of the available computational power. The capacity usage is determined empirically using loop counters embedded in the assembly code.

Chapter 5

Experimental Results

This chapter describes the experimental results obtained with our wideband steerable microphone array, compares the obtained and expected results, and discusses the discrepancies.

5.1 Data Collection

Figure 5-1 shows the setup used to collect test data. The microphone array was placed inside the anechoic chamber at the AT&T Bell Labs Murray Hill facility and attached to a mechanical rotator that is controlled by a MASSComp 6350 system located outside the chamber. Within the chamber a loudspeaker is positioned at a distance of 10 meters (satisfying the requirement that the impinging wave on the microphone array appears planar). The speaker is driven by a set of sinusoidal signals that are generated by the MASSComp. The signals received by the microphone array are appropriately processed (beamformed) and the result is then sent to the MASSComp.

With the loudspeaker, rotator, and microphone array hooked up to the MASSComp the output of the array is determined at a specified angle for a specified frequency. The MASSComp (in conjunction with a workstation) takes the data for small angle increments from 0 degrees to 360 degrees and can thus reconstruct the array's reception pattern in terms of a polar directivity plot. The result is then available to the user in a Matlab file.

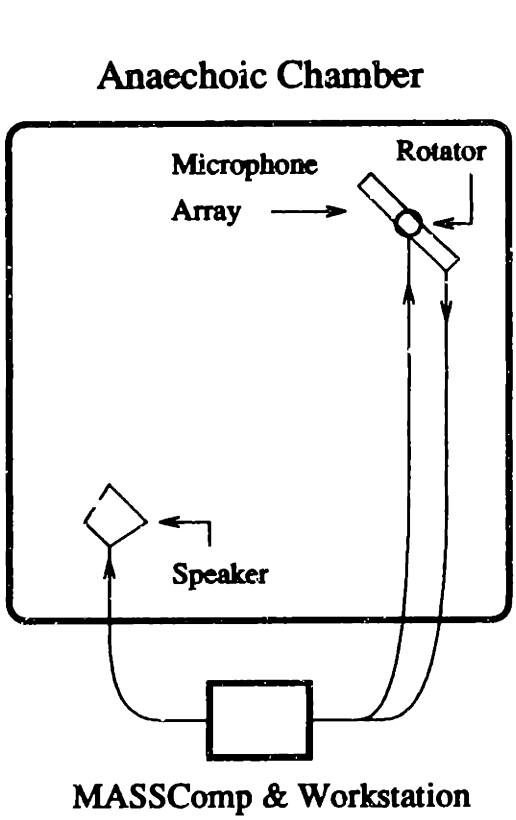

Figure 5-1: Experimental Setup

5.2 Collected Data

Two main sets of data are described in this section. The first data set encompasses the directivity plots for each of the seven beam directions at selected frequencies. The second data set shows the tracking behaviour of the array when set in the automatic steering mode.

Theoretical and measured directivity plots for the seven beam directions and eight test frequencies are shown in figures 5-2 through 5-8. The dashed line indicates the expected directivity plot when using first-order differential microphones ($\cos\theta$ dependance) and the solid line indicates the actual measurements. The $\theta = 0$ direction (broadside) is oriented towards the top of the page.

Figures 5-10 and 5-11 show the results when the microphone array was set to automatic beam steering. Note that the reception area of the microphone array appears as almost the whole scanning region. The reason for this is that as the microphone array was rotated (during measurement), the processing would change beam directions to adjust for the direction where the highest energy was being received. The shape of the directivity plots and

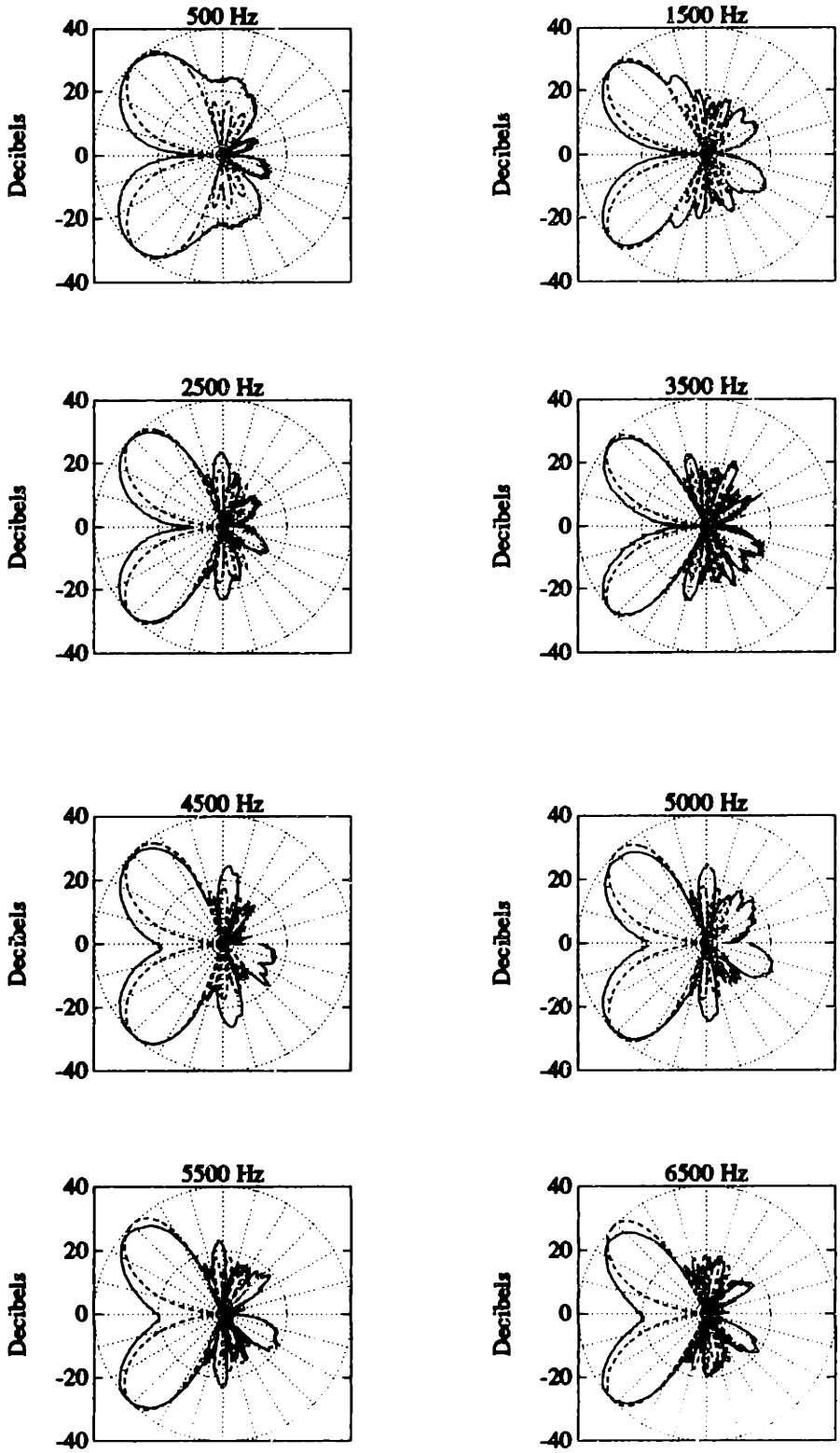

Figure 5-2: Steered beam directivity plot for -54 degrees

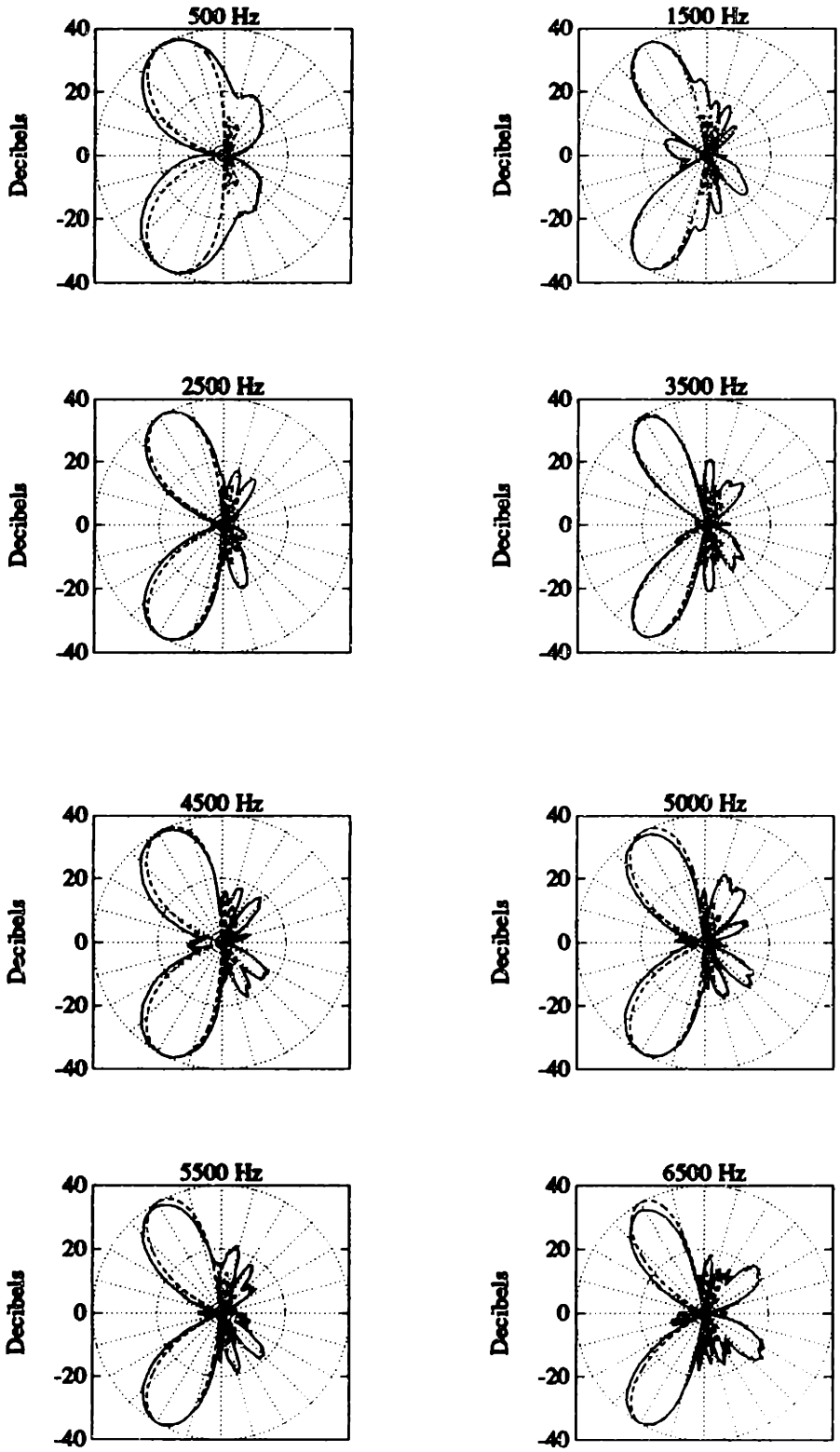

Figure 5-3: Steered beam directivity plot for -32 degrees

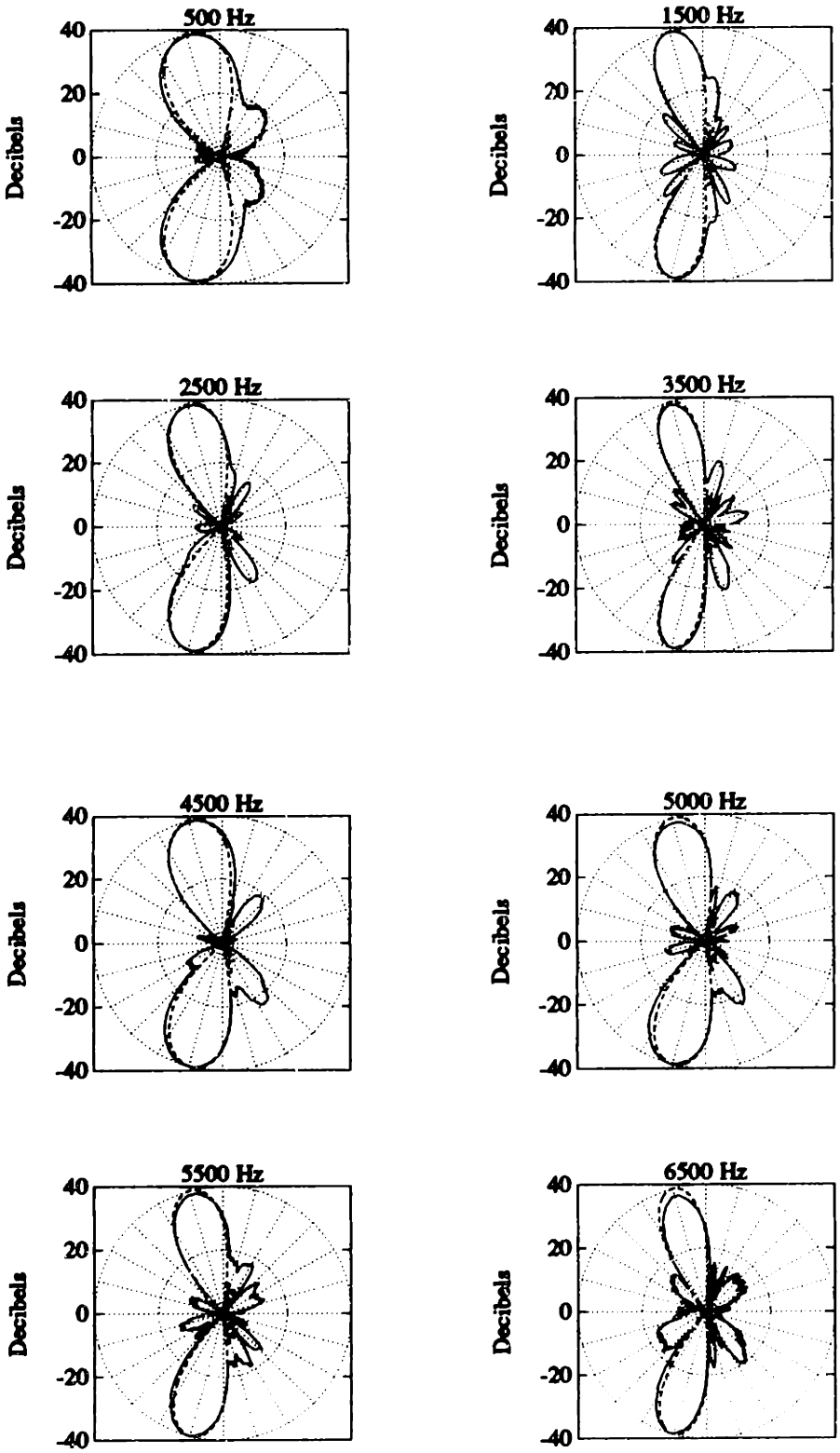

Figure 5-4: Steered beam directivity plot for -16 degrees

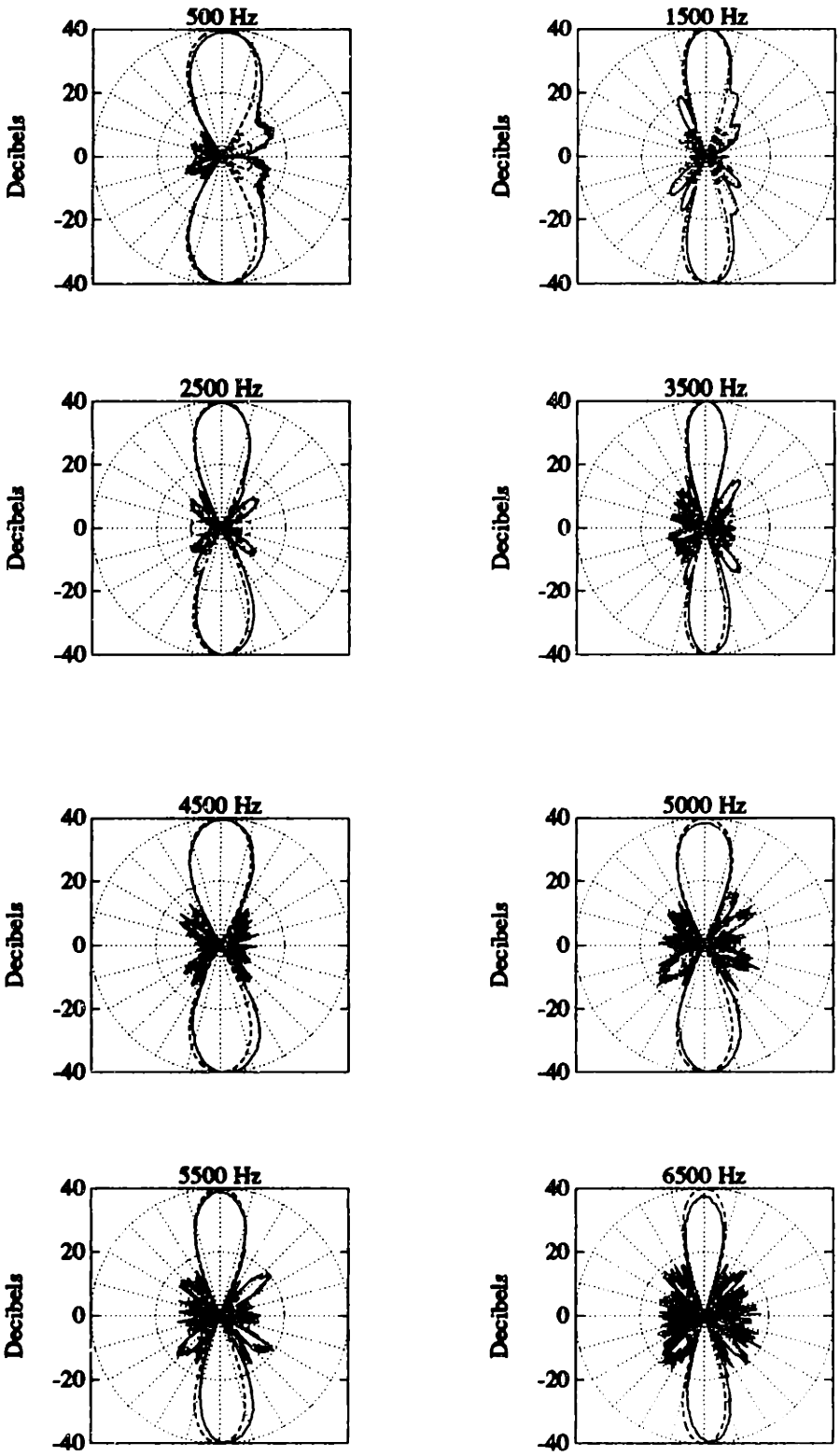

Figure 5-5: Steered beam directivity plot for 0 degrees

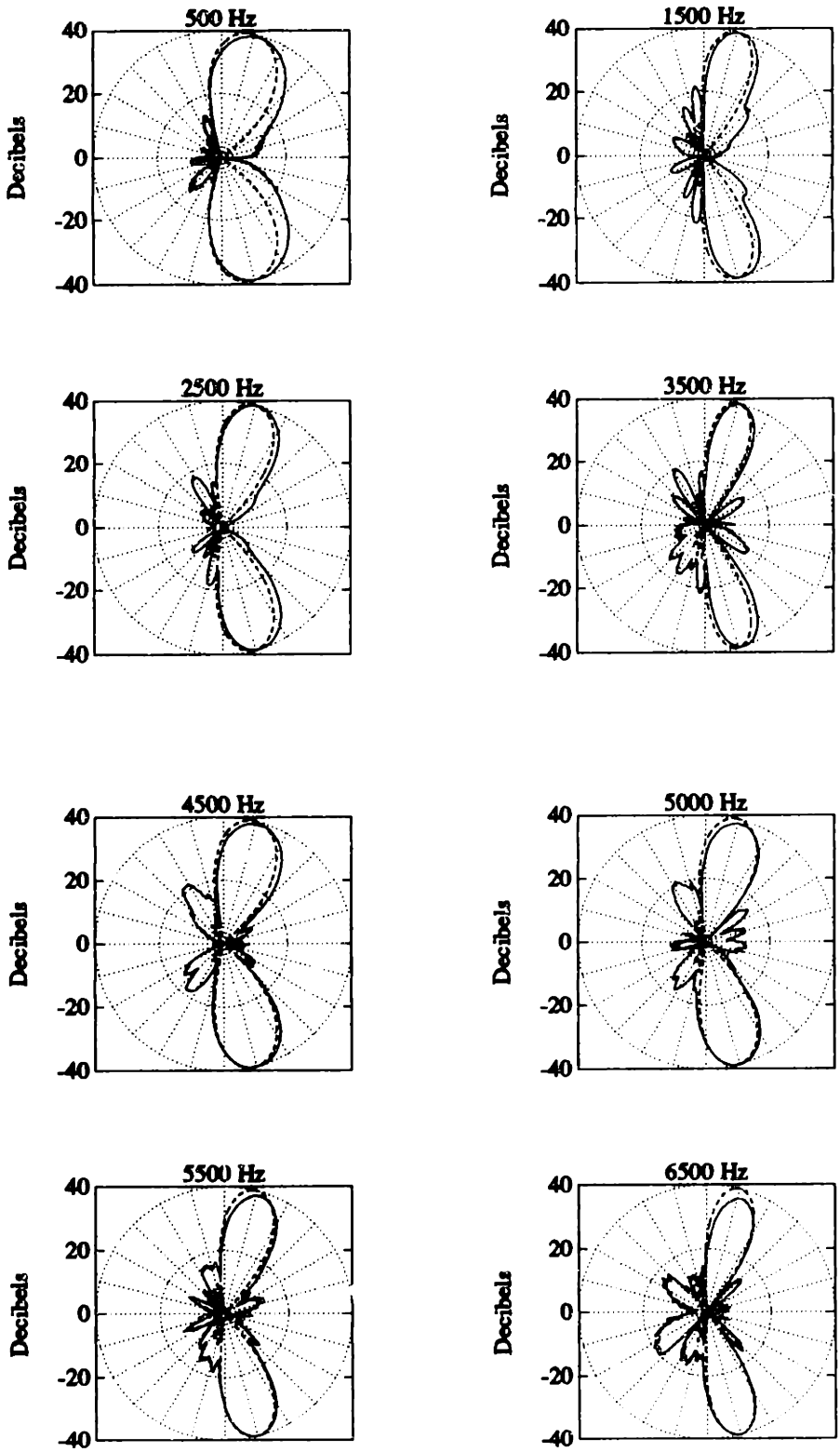

Figure 5-6: Steered beam directivity plot for 16 degrees

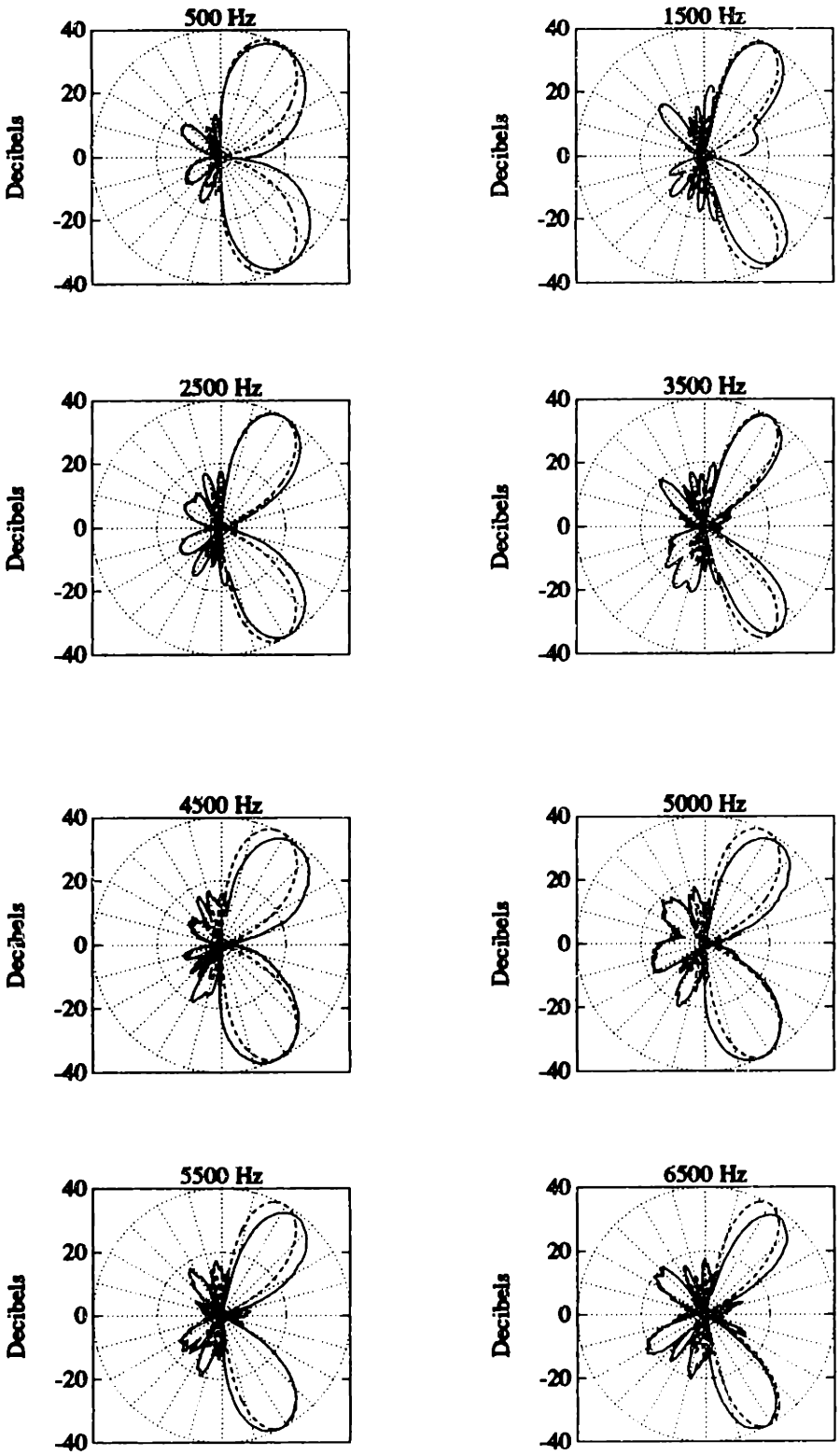

Figure 5-7: Steered beam directivity plot for 32 degrees

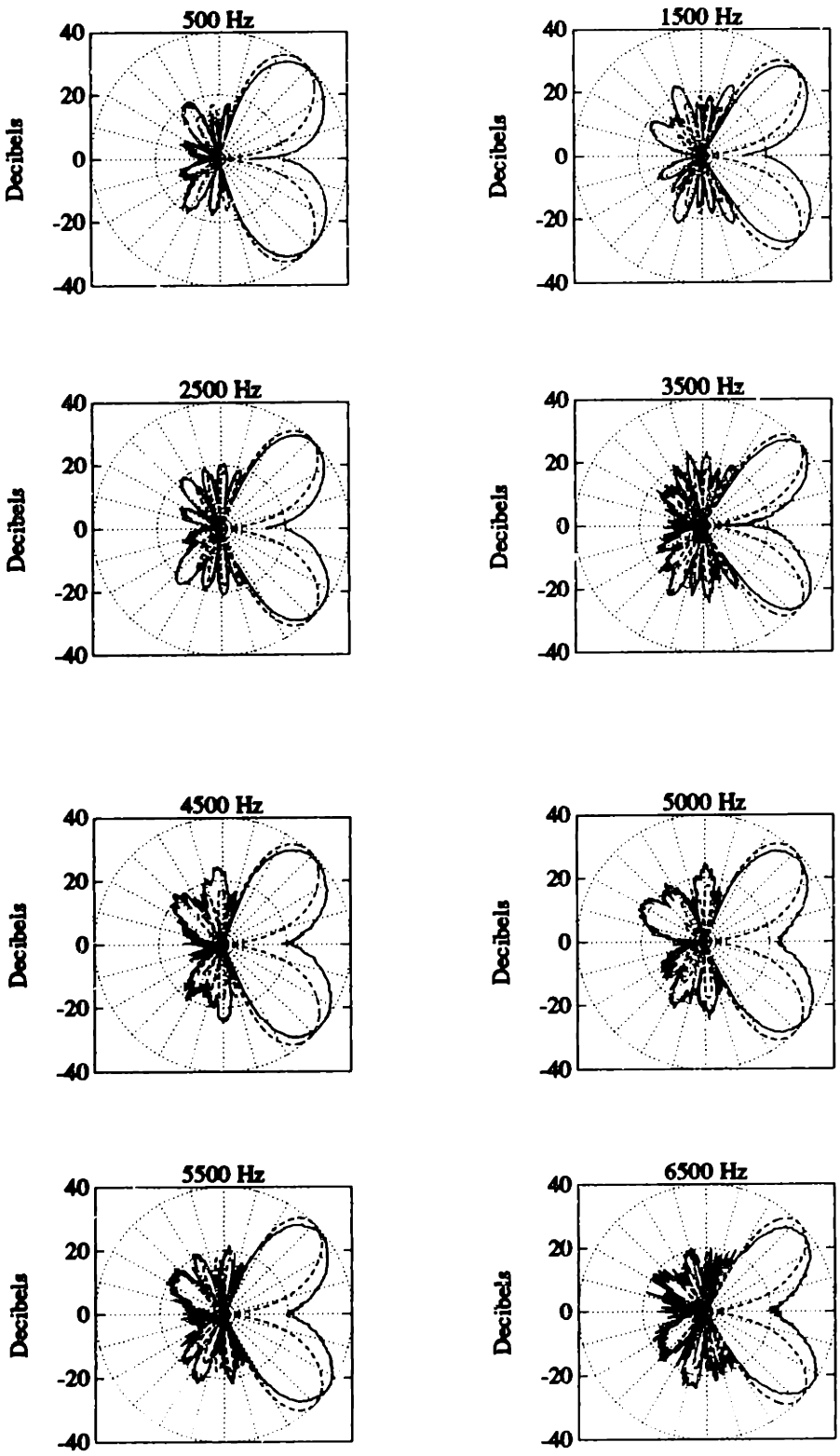

Figure 5-8: Steered beam directivity plot for 54 degrees

the differences between figure 5-10 and 5-11 will be discussed in the next section. Suffice it to say for now that the hysteresis value and the beam coefficient lowpass filters are not the same for both plots.

5.3 Discussion

5.3.1 Directivity Plots For Steered Beams

This discussion focuses on the directivity plots shown in figures 5-2 through 5-8 and refers primarily to the main-beam and sidelobe characteristics.

In figures 5-2 through 5-8, it is clear that the measured main-beam is, for all practical purposes, quite close to the expected beam. Note that as we steer away from broadside, the theoretical and measured results for the main beam start to diverge somewhat. The divergence is exhibited in two ways. The first is that the steering direction between expected and measured beams (see figure 5-2) does not match exactly. The direction mismatch is between 5 and 8 degrees for the eight frequencies exhibited in figures 5-2 and 5-8. This mismatch can be accounted for by imperfections in the microphone array being used. The array is not perfectly straight and, thus, bending in the array contributes to direction mismatch. Furthermore, when setting up the measurements, aligning the array to the source of sound (speaker) to within less than one degree is problematic. The second disparity surfaces with the beamwidth for a steering angle of +/- 54 degrees. The experimental data does not exhibit a null at 90 degrees, while the theoretical results predict a null at 90 degrees from the $\cos\theta$ dependence of the microphones. This fact causes the beamwidth for the steered beam to increase relative to the predicted beamwidth since the predicted $\cos\theta$ attenuation is not fully exhibited. Figures 5-2 and 5-8 illustrate this effect.

The sidelobe characteristics exhibited by the measured results do not match the $\cos\theta$ scaled equiripple characteristic predicted by the theory. By referring first to figure 5-5, it can be seen that both the main beam and the sidelobe attenuation factors are similar (the theoretical sidelobes can be discerned from the figure). The main problem is that the measured sidelobe shapes are not well defined and they exhibit noisy characteristics. Part of this noise can be attributed to system and sampling noise. The system noise

can be accounted in part by imperfect power supply filtering to the first order differential microphones, microphone signal crosstalk through ribbon cable connections, mechanical rotator noise (affects results during the testing stage), and other common causes of analog system degradation. Furthermore, fixtures in the chamber are known to reflect sound and cause aberrations in sidelobe characteristics below -20dB from the peak.

Taking a closer look at the steered beams (off-broadside), another effect can be observed. Comparing the directivity plots for frequencies between 500Hz and 2500Hz for a steering angle of 32 degrees in figure 5-7 with the broadside beam in figure 5-5, note that the sidelobes (which have shifted towards broadside) have grown in amplitude. Recalling that first order differential microphones have a $\cos\theta$ radiation pattern, we introduced a scaling factor of the form $K_{scale} = \frac{1}{\cos\theta_{steered}}$ to compensate for the $\cos\theta$ dependance. Thus, when we beamformed at 32 degrees (figure 5-7), we scaled the beam by $K_{scale} = 1.18$ (1.4dB). Unfortunately, not only did we scale the main beam by K_{scale}, but all the sidelobes as well. Therefore, as the sidelobes move closer to broadside, they receive less attenuation from the $\cos\theta$ dependence, but they are scaled by K_{scale} which is meant to account for attenuation on the off-broadside main beam. This effect becomes more pronounced as the array is steered further from broadside. For example, at 54 degrees (figure 5-8) where $K_{scale} = 1.7$ (4.6dB), the sidelobe growth is much stronger than at 16 degrees (figures 5-4 and 5-6) where $K_{scale} = 1.04$ (.34dB).

Notwithstanding the above (minor) problems, figures 5-2 through 5-8 indicate that the microphone array generally performed very well. Figure 5-9 provides a brief summary of the beamforming results for the array. The expected and measured beamwidths, as measured to the 3dB points of the beam, are given along with a percentage difference. Furthermore, note that due to the symmetric behaviour of the array, only four angles have been presented. Note that the beamwidth mismatch increases for larger steering angles. This is because the expected results accounted for an ideal first order differential microphone with a null at endfire, but the real microphones only approximate the $\cos\theta$ dependance. Nevertheless, the discrepancies exhibited between expected and achieved results only have a small effect on the acoustically perceived system performance.

It should be observed that the results (measured and expected) exhibit a beamwidth

Steering Direction (Degrees)	Frequency (Hz)	$\Delta\theta$ (Degrees)	$\Delta\theta_{\pi}$ (Degrees)	% Difference
0	1500	15.2	16	5%
	2500	18	18.2	1%
	5500	15.9	17.2	8%
16	1500	16	16.2	1%
	2500	18.2	20.2	11%
	5500	16.6	18.2	10%
32	1500	17.2	18.2	6%
	2500	19.2	22.2	16%
	5500	18.2	20.2	11%
54	1500	19.2	21.3	11%
	2500	21.3	25.3	19%
	5500	19.2	23.3	21%

Figure 5-9: Expected vs. Measured Beamwidth

variation as a function of frequency and scan-angle. At broadside (see figure 5-5), the beamwidth variation is evident by comparing the radiation patterns for the 4500Hz and 6500Hz beams. By comparing broadside and steered beams (at a fixed frequency), the scan-angle dependence of the beamwidth can also be observed. Chapter 6 will introduce a method by which these beamwidth variations can be minimized.

5.3.2 Directivity Plot for Automatic Beam Steering

Figures 5-10 and 5-11 show part of the array's behaviour when using automatic beam steering. These plots have the same orientation as the directivity plots in figures 5-2 through 5-8; broadside is defined towards the top of the page. Note that both directivity plots cover the region from -54 degrees to +54 degrees.

These patterns can be understood by recalling the setup used to collect data and how automatic beam steering works (refer to chapter 3). As the array, mounted on a mechanical

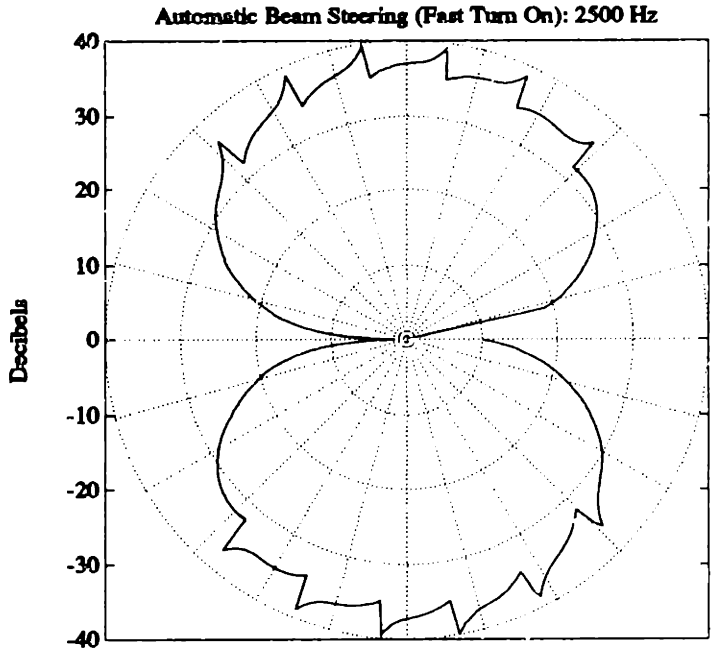

Figure 5-10: Automatic Beam Steering: Fast Turn On

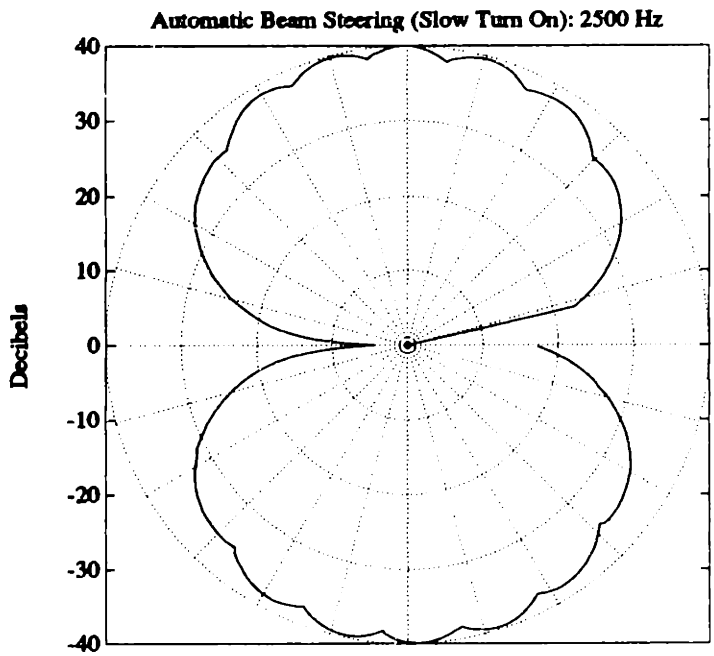

Figure 5-11: Automatic Beam Steering: Slow Turn On

rotator, is turned in a counterclockwise direction, the voting algorithm determines a new beam direction based on the received energy. Thus, if the beam is initially locked on the +54 degrees beam direction, then as the array turns, the beam directed at +32 degrees will next have the highest energy causing the array to switch to that direction. The automatic beam steerer uses three basic parameters to determine the beam direction: energy, hysteresis, and turn-on and turn-off filters. Energy measured in the seven beam directions provides the best first estimate of where to switch directions to. The hysteresis formula is then applied to avoid switching on real and random noise fluctuations. When the direction has been determined, the beamscale coefficients are determined using the turn-on and turn-off lowpass filters. Figure 5-10 shows the effect of using a hysteresis value approximately equal to a 3dB difference between the current beam direction and a possible new direction (this can be expressed mathematically be switching when $E_{new} \geq 1.5E_{current}$). This figure also uses a fast turn-on lowpass filter (10 msecs) and a slow turn-off lowpass filter (1 sec). When the source crosses a beam boundary, the automatic beam steerer decides that the energy of the adjacent steering direction has surpassed that of the current direction by 3dB, thereby causing a beam switch. At the time the beam switches, there is a discontinuity associated with the difference in current and previous energies. The discontinuity has a finite slope introduced by the fast turn-on lowpass filter (the slope would be infinite if there were no turn-on lowpass filter). After the discontinuity, the decay that is observed is determined by a combination of the slow turn-off lowpass filter and the decrease in the previous beam direction energy.

Figure 5-11 shows the radiation pattern during automatic beam steering when the hysteresis value has been reduced to less than 1dB and the turn-on and turn-off transients for the lowpass filters have been equalized (100 msecs). These parameter values result in a smooth radiation pattern.

There is another parameter which was briefly mentioned in chapter 3, the background noise threshold. This value is used to ensure that random beam switching does not occur due to background noise in the room. In its simplest (and implemented) form, this parameter is just a quiescent energy average which is then used to threshold the calculated beam energy for the seven steering directions. Both figures 5-10 and 5-11 exhibit the use of the noise

threshold. Starting at an angle of +90 degrees (i.e. at the right side of the figure) there is no output of the microphone array. This is because the energy received in each of the seven beams is below the noise threshold value and thus no beam is turned on. As soon as we enter the radiation pattern corresponding to one of the seven steering directions (+54 degrees), one of the beams is turned on.

Besides the tests illustrated in figures 5-10 and 5-11, other experiments were also performed that assessed more realistic conditions. One experiment was done by having an individual walk around the room with the beamformer on automatic steering. As the individual walked and talked (rather randomly as most humans do), the array tracked the talker with 100% accuracy in the anechoic chamber. Another experiment used two independent sources set apart from each other. They were then switched on together and their intensities were varied to test the beamformer's ability to discern a predominant source of energy. By setting the hysteresis value to approximately half the maximum received energy, and setting the energy averaging time constant to approximately 1 second, the beamformer tracked changes in the sources' intensities with 95% accuracy in the anechoic chamber.

Chapter 6

Constant Beamwidth Beamforming

As discussed previously, the beamwidth generated by a microphone array varies as a function of frequency and scan angle. Figures 5-2 through 5-8 in chapter 5 illustrate this effect directly from experimental results. Beamwidth variations as a function of frequency and scan angle can be observed within each frequency band.

Several methods have been suggested on minimizing beamwidth variations as a function of frequency [14, 15]; they are summarized in [4]. We will describe a variation on the above methods as presented in [4, 12]. The suggested method minimizes beamwidth variations as a function of frequency by performing elemental lowpass filtering (i.e. lowpass filtering array element outputs). Beamwidth variations as a function of scan angle can then be minimized by changing the elemental lowpass filters. Beamwidth variations as a function of scan angle may also be minimized by explicitly changing the array shading coefficients.

6.1 Constant Beamwidth as a Function of Frequency

Beamwidth is inversely proportional to the array length (see equations 2.10 and 2.11 in Chapter 2). Although the array has been broken up into subbands to reduce beamwidth variations as a function of frequency, the beamwidth still varies by about a factor of two between low and high frequencies in one subband. If the effective length of the array

can be shortened, the beamwidth at high frequencies can be increased. One way this can be accomplished is by performing elemental lowpass filtering. The outer microphones in the subarray use progressively lower cutoff frequency lowpass filters relative to those microphones close to the center. This reduces the effective length of the array as the frequency increases since the outer microphone outputs are attenuated at high frequencies.

Besides maintaining the beamwidth constant as a function of frequency, we can also keep the beamwidth constant as a function of scan angle. This is accomplished by determining a new set of elemental lowpass filters whose resulting steered beam matches the beamwidth of the broadside beam. This means that each steered beam will have a different set of elemental lowpass filters.

6.1.1 Mathematical Derivation of Elemental Lowpass Filters

The derivation of the elemental lowpass filters stems from the argument that one can achieve constant beamwidth over a wide frequency range by forming multiple beams steered at incremental angles from each other. This method suggests that rather than forming one beam, L beams should be formed. We then control the intermicrophone delay τ for each beam such that we maintain the zero of the outermost beam constant. This is equivalent to controlling the steering direction of the marginally steered beams such that as we increase frequency (and the beamwidth shrinks), we steer the marginal beams outwards. Figure 6-1 illustrates multi-beamforming as a way to keep the beamwidth constant as a function of frequency.

Deriving the elemental lowpass filters requires solving for the internull beamwidth of a steered beam. Expressions for the internull beamwidth using uniform and Chebyshev excitation are presented here. From Chapter 2, the internull beamwidth when using uniform excitation is:

$$\triangle\theta_{rect} = 2\sin^{-1}(\frac{2\pi}{M\omega\tau_o}) \tag{6.1}$$

The internull beamwidth when using Chebyshev excitation is:

$$\triangle\theta_{cheb} = 2\sin^{-1}\{\frac{2}{\omega\tau_o}cos^{-1}(\frac{1}{x_o}\cos(\frac{\pi}{4N}))\} \tag{6.2}$$

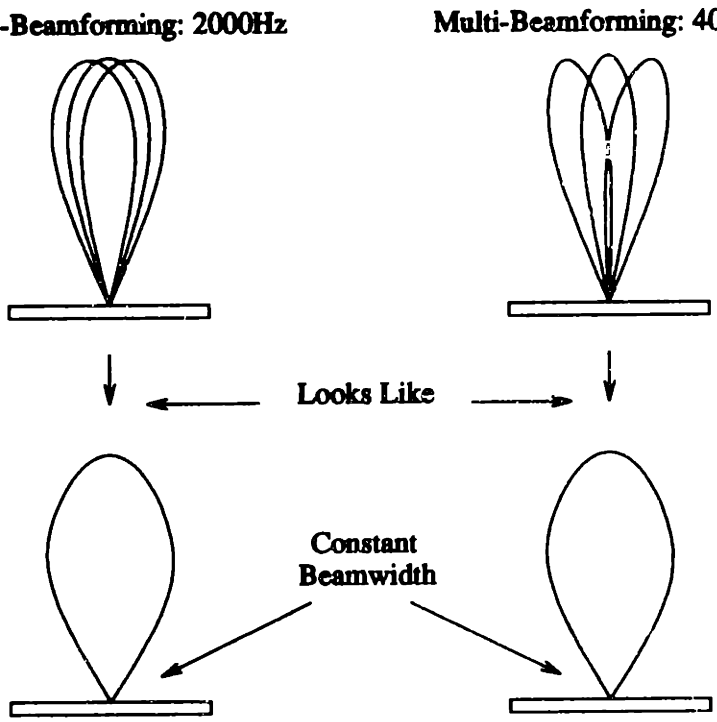

Figure 6-1: Multi-beamforming

where $\tau_o = \frac{d}{c}$, M equals the number of elements in the array, $N = \frac{M-1}{2}$, and $x_o = \frac{1}{M}\cosh^{-1}(R)$ where R is the ratio of the main beam amplitude to sidelobe amplitude.

The internull beamwidth of a steered beam can be determined by solving equation 2.7 in Chapter 2 for a given set of $\{a_n\}$. Given a uniformly weighted microphone array, the internull steered beam beamwidth is:

$$\Delta\theta_{rect} = \theta_A - \theta_B \tag{6.3}$$

$$\theta_A = \sin^{-1}\left(\frac{2\pi}{M\omega\tau_o} + \sin\theta_o\right) \qquad \theta_B = \sin^{-1}\left(\frac{-2\pi}{M\omega\tau_o} + \sin\theta_o\right)$$

where θ_o defines the steering angle. The elemental lowpass filters required for a uniformly weighted microphone array can now be determined.

Forming L marginally steered beams, the overall beam beamwidth is found to be twice the angle at which the outermost null is found (null flanking the main beam). From equation

6.3 twice the outermost null gives:

$$\Delta\theta_{rect} = 2\sin^{-1}(\frac{2\pi}{M\omega\tau_o} + \sin\theta_o) \tag{6.4}$$

Letting the outermost steering angle θ_o vary as a function of frequency, θ_o can be defined as $\theta_o(\omega) = \sin^{-1}(\frac{\tau(\omega)}{\tau_o})$ where $\tau_o = \frac{d}{c}$. This is done by recalling that a steered beam was originally obtained by scaling the microphones by a progressive exponential whose delay determines the steering angle. In this case, the delay τ from Chapter 2 is made a function of frequency. Substituting for θ_o in equation 6.4:

$$\Delta\theta_{rect} = 2\sin^{-1}(\frac{2\pi}{M\omega\tau_o} + \frac{\tau(\omega)}{\tau_o}) \tag{6.5}$$

Equation 6.5 above can be solved for $\tau(\omega)$ while holding $\Delta\theta_{rect}$ constant. The phase function $f(\omega)$ can then be found by solving $f(\omega) = -\omega\tau(\omega)$:

$$f(\omega) = \frac{2\pi}{M}(1 - \frac{\omega}{\omega_o}) \tag{6.6}$$

where $\omega_o = \frac{2\pi}{K\tau_o M}$ and K is the arcsine of the desired beamwidth.

The phase function for a a Chebyshev window can be similarly solved for by first observing that the beamwidth of a steered beam is:

$$\Delta\theta_{cheb} = 2\sin^{-1}\{\frac{2}{\omega\tau_o}\cos^{-1}(\frac{1}{x_o}\cos(\frac{\pi}{4N})) + \sin\theta_o\} \tag{6.7}$$

Following the same approach used above by substituting $\sin\theta_o$ with $\frac{\tau(\omega)}{\tau_o}$, the phase function for a Chebyshev window is found to be:

$$f(\omega) = 2\{\cos^{-1}(\frac{1}{x_o}\cos(\frac{\pi}{4N}))\}(1 - \frac{\omega}{\omega_o}) \tag{6.8}$$

where $\omega_o = \frac{2\alpha}{\tau_o K}$, $\alpha = \cos^{-1}(\frac{1}{x_o}\cos(\frac{\pi}{4N}))$, and K is the arcsine of the desired beamwidth.

Having thus determined the phase function $\tau(\omega)$, $L = 2r + 1$ beams can be formed at incremental angles where the steering is determined by a progressive exponential with $\tau(\omega)$ as the frequency dependent delay. The L formed beams are added to obtain the following

spatial response:

$$H_{cb}(\omega,\theta) = \sum_{i=-r}^{r} \sum_{n=-N}^{N} a_n e^{-jn\omega\tau_o \sin\theta} e^{-jn\omega\tau(\omega)\beta(i,r)} \tag{6.9}$$

where $\beta(i,r)$ determines the relative location of the steered beams. Using a symmetrical set of a_n equation 6.9 can be redefined as:

$$H_{cb}(\omega,\theta) = \sum_{n=-N}^{N} B_n(\omega) e^{-\frac{jn\omega d \sin\theta}{c}} \tag{6.10}$$

where

$$B_n(\omega) = a_n\{1 + \sum_{i=1}^{r} 2\cos(nf(\omega)\beta(i,r))\} \tag{6.11}$$

Therefore, equation 6.11 describes the elemental lowpass filters needed to obtain a constant beamwidth over a wide frequency range. The function $\beta(i,r)$ can be used to space the steered beams at optimal locations. In the simplest scenario, $\beta(i,r) = \frac{i}{r}$.

6.1.2 Elemental Lowpass Filter Design

The previous section dealt only with the derivation of the elemental lowpass filters needed to achieve constant beamwidth as a function of frequency. This section will describe the actual design of these filters. The design involves finding $f(\omega)$ in equation 6.11, determining the desired beamwidth, and determining the number of marginally steered beams. A temporal response can then be determined with one of many filter design techniques [8, 19].

To illustrate the issues in elemental lowpass filter design, we will construct the necessary lowpass filters to achieve a constant beamwidth of 58 degrees using Chebyshev excitation. The desired beamwidth matches the beamwidth of a broadside beam at a frequency $\frac{c}{4d}$ (c =speed of sound in medium and d =interelement spacing) in our harmonically nested microphone array. By letting $\beta(i,r) = \frac{i}{r}$, r=3, d =.02m, and using 30dB sidelobe attenuation, the elemental lowpass filters can be fully determined.

With the above parameters, the phase function can be found and consequently the elemental lowpass filters $B_n(\omega)$. Figure 6.1.2 shows the magnitude of the elemental lowpass

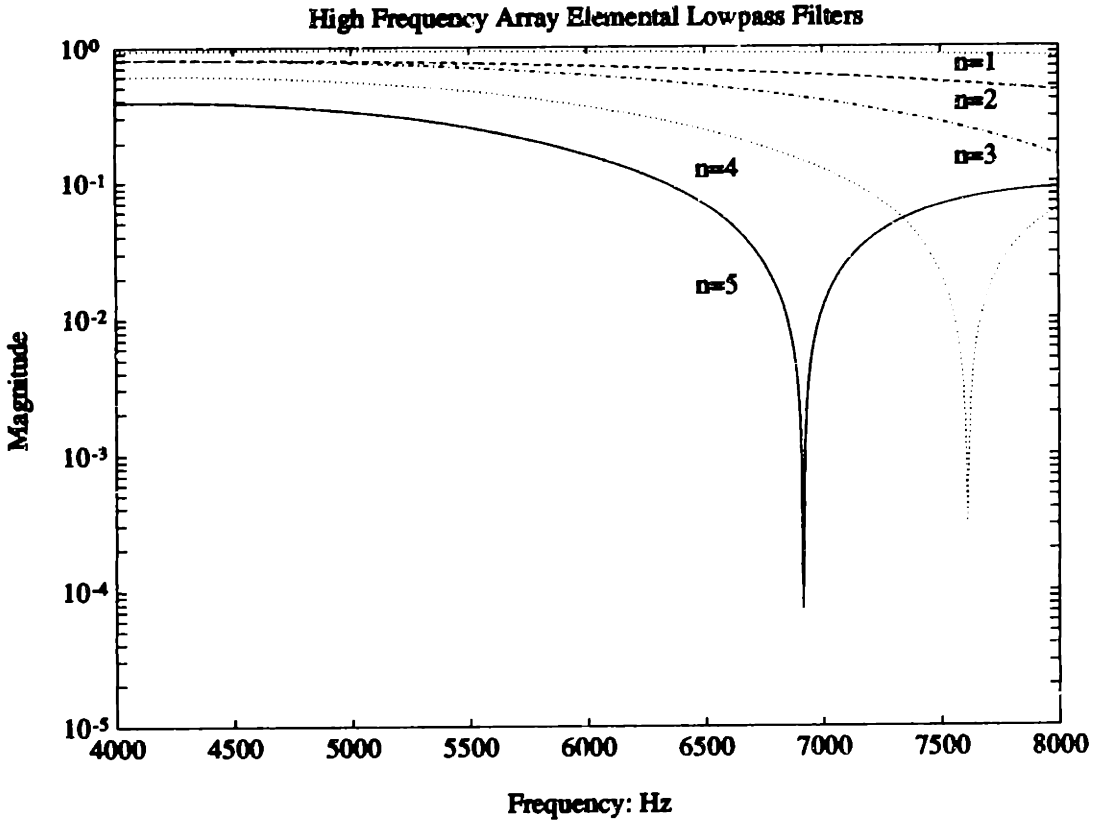

Figure 6-2: Elemental Lowpass Filter with 30dB Sidelobe Attenuation

filters for the nth microphone relative to the center element. Notice as n increases (moving away from the center of the array), the elemental lowpass filters have lower frequency cutoffs.

6.2 Constant Beamwidth as a Function of Scan Angle

The design of the elemental lowpass filters in the previous section focused on keeping the beamwidth constant as a function of frequency. Yet is was shown earlier that the beamwidth not only varies with frequency, but also with the scan angle. Once again, applying the concept of elemental lowpass filtering, a method for keeping the beamwidth constant as a function of scan angle and beamwidth can be derived.

To achieve constant beamwidth as simultaneous functions of scan angle and frequency, we need to design Q elemental lowpass filters, one for each scanning direction (for seven beam directions $Q = 7$). The lowpass filters will be designed for a broadside beam. The main

design issue is to determine what broadside beamwidth will yield the desired beamwidth when steered. For example, suppose we were to form three beams, one at broadside and two at forty-five degree angles. We would solve for the elemental lowpass filters by keeping the beamwidth constant at a predetermined value w. The elemental lowpass filters for the broadside beam would be designed for a beamwidth of w. The elemental lowpass filters for the two steered beams would be designed for a broadside beamwidth of $w - \delta$ such that when steered, the resulting beamwidth will be approximately w.

A method that provides a good first-order approximation to δ above is the following. The value δ can be determined by first finding the broadside beamwidth of an array with microphone weights $\{a_n\}$ (the beamwidth corresponds to the single-beam beamwidth discussed in Chapter 2); call this beamwidth $\triangle\theta_a$. The beamwidth of a steered beam can then be determined when using the same set of weights; call this beamwidth $\triangle\theta_b$. With both of these beamwidths, we can define a ratio ξ which approximates the beamwidth variation between broadside and steered beam:

$$\xi = \frac{\triangle\theta_a}{\triangle\theta_b} \tag{6.12}$$

Once ξ has been determined, δ can be approximated by $\delta = w(1 - \xi)$. The beamwidth w is determined by the application. Therefore, the elemental lowpass filters for the steered beams would solve for a broadside beamwidth of ξw.

6.3 Dynamic Beamwidth Control

In addition to keeping the beamwidth constant as a function of frequency and scan angle, it may be of use for certain applications to be able to control the beamwidth dynamically. This would entail allowing user control of beamwidth depending on the specific scenario. Controlling the beam could involve zooming (decreasing the beamwidth) or expanding (increasing the beamwidth).

Following the above discussion, one could use the elemental lowpass filter method as a means to control beamwidth modulation. Specifically, D elemental lowpass filters could be designed, one for each desired beamwidth. The filters are designed using only one

set of $\{a_n\}$. These filters could then be dynamically interchanged or interpolated in the beamformer under user control. If it was desired to control the beamwidth in 10 degree increments from 50 degrees to 100 degrees, a set of elemental lowpass filters could be found for 50, 60, 70 , 80, 90, and 100 degrees. These filters could then be dynamically downloaded to the beamformer.

Another approach to modulate the beamwidth would be to determine D sets of $\{a_n\}$, one for each desired beamwidth. These could then be downloaded to the beamformer when the user so desires. Furthermore, a set of elemental lowpass filters can then be determined for each of $\{a_n\}$ to ensure constant beamwidth beamforming as a function of frequency. Determining a new set of $\{a_n\}$ for each beamwidth is approximately equivalent to designing elemental lowpass filters for each beamwidth. One should note that designing elemental lowpass filters does in fact change the shading coefficients.

Chapter 7

Conclusion

7.1 Summary

A wide-band digital beamforming microphone array has been designed and implemented. The directivity results obtained when using manual beam steering closely matched those expected from theory. Furthermore, automatic beam steering operation performed well and tracked sound sources with high accuracy. When testing automatic steering with one sound source, the beamformer tracked the motion of the source with 100% accuracy in an anechoic chamber. When testing automatic steering with two sources, the beamformer was able to discern the predominant source of energy with approximately 95% accuracy (after proper adjustment of the hysteresis value) in an anechoic chamber. The utility of the beamformer is further enhanced by a flexible user interface which gives access to beamforming algorithm parameters and provides good system trouble-shooting diagnostics.

The out-of-band (spatial) attenuation exhibited by the beamformer was excellent, potentially helping to break the feedback loop problems discussed in the introduction. By proper placement of the beamformer in a teleconferencing room, the speaker output can be removed from the feedback path by ensuring that it's direct field falls outside the main beam. Such an application greatly enhances the acoustic quality in a teleconferencing situation. Beyond ensuring that feedback-loop effects are minimized, proper placing of the microphone array can also help reduce detrimental effects associated with the room's reverberant field. Specifically, by focusing the main beam on the talker, the beamformer

enhances the direct signal level over the reverberant field received by the array.

As mentioned previously, the automatic beam steering capabilities worked very well. With careful selection of the beamforming parameters, the automatic beamforming algorithm yielded very predictable results in determining the correct source of sound when more than one source of sound was active. Due to the flexibility built into the beamformer, the voting algorithm can be easily modified so that the beamformer can track two speakers at the same time (i.e. keep two beams on).

The system was also designed to ensure that improved system performance is easily achieved with minor coding changes. Constant beamwidth beamforming and dynamic beamwidth control are easily incorporated into the current system. Once again, the designed flexibility of the system allows improved performance with no hardware costs (although there is a limit on the system's compute power).

7.2 Future Directions

The current beamformer provides an excellent platform for further improvements in microphone array technology. Immediate issues that could be studied with the beamformer are improved automatic beam steering capabilities to achieve 100% accuracy in discriminating between two sound sources. Automatic steering can also be enhanced by altering the voting algorithm to track two talkers simultaneously and appropriately scale the beamscale coefficients to account for a second talker (instead of only having 1 and .1 as beamscale coefficients, a .5 beamscale coefficient may be introduced). Implementation of the constant beamwidth beamforming ideas should also be incorporated. Shortly following this, dynamic beamwidth control could also be studied on the current platform using chapter 6 as a guideline.

By incorporating dynamic beamwidth control capabilities, a new range of applications for the beamformer appear. By using the beamformer in conjunction with a camera, acoustic zooming could track visual zooming by the camera. Such an application could be used in conference halls and with television crews.

Beyond the teleconferencing and television applications, many areas could benefit by using an acoustic beamformer. Beamformer performance can be optimized to make speech

recognizers that are more robust. A major problem associated with automatic speech recognition is distinguishing noise from signal. Close-talking microphones and other expensive equipment are usually required to ensure high quality speech recognition. By adapting array technology to this arena, high quality recognizers could be developed for noisy environments (airports, train stations, etc.).

Array technology could also be used to improve acoustic echo-cancellation. As was shown with the current system implementation, tracking a talker around a room becomes a realistic goal. By tracking a talker, an acoustic beamformer can thus help ensure high directivity in the proper direction and help remove far-end speech present in the reverberant field. With this added directivity, echo-cancellers could then yield much higher performance.

Bibliography

[1] J.E. Byrne. "The Virtual Corporation". *BusinessWeek.* Feb. 8, 1993. pp 98-102.

[2] "Autodirective microphone systems" by J.L. Flanagan, D. A. Berkley, G. W. Elko, and J. E. West. *Acustica.* 73:58-71, February 1991.

[3] W. Kellerman. Design and Implementation of a Digitally Steered Microphone Array. Unpublished Paper. April 22, 1991.

[4] Michael Goodwin. Implementation and Applications of Electroacoustic Array Beamformers. Massachusetts Institute of Technology. June 1992. Bachelor of Science and Master of Science Thesis.

[5] AT&T DSP3210 Information Manual. AT&T Microelectronics.

[6] B. D. Steinberg. *Principles of Aperture & Array System Design.* John Wiley & Sons, 1976.

[7] L. C. Shen and J.A. Kong. *Applied Electromagnetism.* PWS Publishers, Second Edition, 1987.

[8] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[9] W. M. Siebert. *Circuits, Signals, and Systems.* M.I.T. Press, Fourth Printing, 1989.

[10] C.L. Dolph. A current distribution for broadside arrays which optimizes the relationship between width and side-lobe level. *Proceedings of the I.R.E. and Waves and Electrons.* June, 1946.

[11] Noise-Cancelling Back Electret Condenser Microphone Cartridge - WM-55D103. Panasonic/Matsushita Communication Industrial Co., Ltd. Secaucus NJ.

[12] "Constant Beamwidth Beamforming", M. M. Goodwin and G. Elko. *Proceeding of 1993 IEEE ICASSP*, pp. I-169-I-772.

[13] MP3210 User's Manual. Ariel Corporation.

[14] E.L. Hixson and K.T. Au. Broadband constant beamwidth acoustical arrays. Technical report, Technical Memorandum No. 19, Acoustics Research Laboratory, University of Texas at Austin, 1970.

[15] R.P. Smith Constant Beamwidth receiving arrays for broad band sonar systems. *Acustica*, Vol. 23,pp21-26, 1970.

[16] H.F. Olson. *Acoustical Engineering.* D. Van Nostrand Company Inc., 1957.

[17] High Speed Multiplexor designed at AT&T Bell Laboratories, Murray Hill.

[18] G. Oetken et al. New results in the design of digital interpola tors. *IEEE Transactions on Acoustics, Speech, and Signal Processing.* Vol. ASSP-23, No.3,pp301-308, June1975.

[19] J.N. Little and L. Shure. *Signal Proceesing Toolbox for use with Matlab.* The Mathworks Inc., August 1988.