2009

Enhancements to the Generalized Sidelobe Canceller for Audio Beamforming in an Immersive Environment

Phil Townsend
University of Kentucky, jptown0@engr.uky.edu

ABSTRACT OF THESIS

Enhancements to the Generalized Sidelobe Canceller for Audio Beamforming in an Immersive Environment

The Generalized Sidelobe Canceller is an adaptive algorithm for optimally estimating the parameters for beamforming, the signal processing technique of combining data from an array of sensors to improve SNR at a point in space. This work focuses on the algorithm's application to widely-separated microphone arrays with irregular distributions used for human voice capture. Methods are presented for improving the performance of the algorithm's blocking matrix, a stage that creates a noise reference for elimination, by proposing a stochastic model for amplitude correction and enhanced use of cross correlation for phase correction and time-difference of arrival estimation via a correlation coefficient threshold. This correlation technique is also applied to a multilateration algorithm for an efficient method of explicit target tracking. In addition, the underlying microphone array geometry is studied with parameters and guidelines for evaluation proposed. Finally, an analysis of the stability of the system is performed with respect to its adaptation parameters.

Multimedia Elements Used: WAV (.wav)

KEYWORDS: Beamforming, Digital Signal Processing, Microphone Arrays, Audio Signal Processing, Stochastics

Author's signature: Phil Townsend

Date: December 15, 2009

Enhancements to the Generalized Sidelobe Canceller for Audio Beamforming in an Immersive Environment

By
Phil Townsend

Director of Thesis: Kevin D. Donohue

Director of Graduate Studies: Stephen Gedney

Date: December 15, 2009

RULES FOR THE USE OF THESES

Unpublished theses submitted for the Masters degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name	Date

THESIS

Phil Townsend

The Graduate School
University of Kentucky
2009

Enhancements to the Generalized Sidelobe Canceller for Audio Beamforming in an Immersive Environment

---

THESIS

A thesis submitted in partial
fulfillment of the requirements for
the degree of Master of Science in
Electrical Engineering in the College
of Engineering at the University of
Kentucky

By
Phil Townsend
Lexington, Kentucky

Director: Dr. Kevin D. Donohue, Professor of Electrical and Computer Engineering
Lexington, Kentucky 2009

To my loving parents Ralf and Catherine, as well as my close friends Em, Katie, Robert, and Richard.

ACKNOWLEDGMENTS

First and foremost I'm deeply thankful to my advisor, Dr. Kevin Donohue, for his support during my work at the University of Kentucky. His patient guidance as both professor and mentor through the thesis process and during the several classes he's taught me throughout my academic career has been exceptional.

I'd like to thank everyone at the UK Vis Center for their support and discussion of our audio work and in particular Drs. Jens Hannemann and Samson Cheung for agreeing to take the time to sit for my defense committee.

And finally, I'd like to thank all my of my family members and closest friends for their love and support during my college career as it's led me to the completion of this thesis and beyond.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FILES

Clicking on the file name will play the selected WAV file in your environment's default audio player.

1. Amplitude Correction Sound Files (Chapter 2)
    a) Linear Array
        i. Target Speaker Alone: *target.wav* (1.1 MB)
        ii. Cocktail Party Closest Mic: *closestMic.wav* (1.1 MB)
        iii. Traditional GJBF Overall Output: *yStandard.wav* (1.1 MB)
        iv. 1/r Model Overall Output: *y1r.wav* (1.1 MB)
        v. ISO Model Overall Output: *yIso.wav* (1.1 MB)
        vi. Statistical Model Overall Output: *yStat.wav* (1.1 MB)
        vii. Perfect BM Overall Output: *yPerfect.wav* (1.1 MB)
    b) Perimeter Array
        i. Target Speaker Alone: *target.wav* (1.1 MB)
        ii. Cocktail Party Closest Mic: *closestMic.wav* (1.1 MB)
        iii. Traditional GJBF Overall Output: *yStandard.wav* (1.1 MB)
        iv. 1/r Model Overall Output: *y1r.wav* (1.1 MB)
        v. ISO Model Overall Output: *yIso.wav* (1.1 MB)
        vi. Statistical Model Overall Output: *yStat.wav* (1.1 MB)
        vii. Perfect BM Overall Output: *yPerfect.wav* (1.1 MB)
2. Cross Correlation Sound Files for Linear Array (Chapter 3)
    a) $\rho_{thresh} = .1$: *y1.wav* (1.1 MB)
    b) $\rho_{thresh} = .5$: *y5.wav* (1.1 MB)
    c) $\rho_{thresh} = .9$: *y9.wav* (1.1 MB)

Chapter 1

Introduction and Literature Review

1.1 A Brief History and Motivation for Study

Beamforming is a spatial filtering technique that isolates sound sources based on their positions in space [1]. The technique originated in radio astronomy during the 1950's as a way of combining antenna information from collections of antenna dishes, but by the 1970's beamforming began to be explored as a generalized signal processing technique for any application involving spatially-distributed sensors. Examples of this expansion include sonar, to allow submarines greater ability to detect enemy ships using hydrophones, or in geology, enhancing the ability of ground sensors to detect and locate tectonic plate shifts [2].

It was around this time that microphone array beamforming in particular became an active area of research, where the practice amounts to placing a virtual microphone at some position without physical sensor movement. Applications of audio beamforming include hands-free listening and tracking of sound sources for notetaking in an office environment, issuing verbal commands to a computer, or surveillance with a hidden array. In the present day the implementation cost of an array is low enough to be a feasible technology for the consumer market. In fact, some common PC software packages currently support small scale arrays such as Microsoft Windows Vista [3].

The present state of the art has seem some ability to improve acoustic SNR (signal to noise ratio) through the use of a microphone array but the performance still leaves much to be desired, especially under poor SNR conditions [2]. It is currently believed that nonlinear techniques, such as the adaptive Generalized Sidelobe Canceller (GSC), will likely provide the most benefits given further study. Hence the study of the GSC, along with several attempts to improve its performance at enhancing human voice capture, will be the focus of this work. In particular, we'll study what's referred to as the cocktail party problem, where we attempt to pull a human voice at one spatial location out of an acoustic scene that has several competing human voices at different locations.

1.2 The Basics of Beamforming

1.2.1 A Continuous Aperture

The concept of a beamformer is derived from the study of a theoretical continuous aperture (a spatial region that transmits or receives propagating waves) and modeling a microphone array as a sampled version at discrete points in space. The technique can be briefly formulated by first expressing the signal received by the aperture as the application of a linear filter to some wave at all points along the aperture via the convolution [4]

$$x_R(t, \mathbf{r}) = \int_{-\infty}^{\infty} x(\tau, \mathbf{r}) a(t - \tau, \mathbf{r}) d\tau \tag{1.1}$$

where $x(t, \mathbf{r})$ is the signal at time t and spatial location $\mathbf{r}$ and $a(t, \mathbf{r})$ is the impulse response of the receiving aperture at t and $\mathbf{r}$. Equivalently, the Fourier transform of (1.1) yields the frequency domain representation

$$X_R(f, \mathbf{r}) = X(f, \mathbf{r}) A(f, \mathbf{r}) \tag{1.2}$$

where $A(f, \mathbf{r})$ is called the *aperture function*, as it describes the sensitivity of the receiving aperture as a function of frequency and position along the array. It can be shown that the *far field directivity pattern*, or *beampattern*, which describes the received signal as a function of position in space for sources significantly distant from the array (Fresnel number $F << 1$), is the Fourier transform of the aperture function

$$D(f, \boldsymbol{\alpha}) = \mathcal{F}\{A(f, \mathbf{r})\} = \int_{-\infty}^{\infty} A(f, \mathbf{r}) e^{j2\pi \boldsymbol{\alpha} \cdot \mathbf{r}} d\mathbf{r} \tag{1.3}$$

where $\boldsymbol{\alpha}$ is the three-element direction vector of a wave in spherical coordinates

$$\begin{aligned} \boldsymbol{\alpha} &= \frac{1}{\lambda} [\, \sin\theta \cos\phi \quad \sin\theta \sin\phi \quad \cos\theta \,] \\ &= [\, \alpha_x \quad \alpha_y \quad \alpha_z \,] \end{aligned} \tag{1.4}$$

with θ the zenith angle, ϕ the azimuth angle, λ the sound source wavelength and the elements of the vector corresponding to the x, y, and z Cartesian directions, respectively.

1.2.2 The Delay-Sum Beamformer

The Delay-Sum Beamformer (DSB) is the simplest of the beamforming algorithms and follows closely from the above discussion of a continuous aperture. The DSB arises when one transforms the integration in (1.3) to a summation over a discrete number of microphones and models the aperture function as a set of complex weights w_n that may be chosen freely for each microphone.

$$D(f, \boldsymbol{\alpha}) = \sum_{n=1}^{M} w_n(f) e^{j2\pi \boldsymbol{\alpha} \cdot \mathbf{r}_n} \tag{1.5}$$

where M is the number of microphones in the array. If one chooses w_n as a set of purely phase terms the beamfield shape will be maintained [1] but its peak will shift, where if

$$w_n(f) = e^{-j2\pi\boldsymbol{\alpha}'\cdot\mathbf{r}_n}$$

then

$$D'(f,\boldsymbol{\alpha}) = \sum_{n=1}^{M} e^{j2\pi(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\cdot\mathbf{r}_n} = D(f,\boldsymbol{\alpha}-\boldsymbol{\alpha}') \tag{1.6}$$

This choice of phase terms in the frequency domain corresponds to delays in the time domain, and for the DSB these delays are taken as the time a sound wave requires to propagate from the Cartesian position of its source (x_s, y_s, z_s) to the n^{th} microphone at (x_n, y_n, z_n), which one may express as

$$\tau_n = \frac{d_n}{c} = \frac{\sqrt{(x_s-x_n)^2+(y_s-y_n)^2-(z_s-z_n)^2}}{c} \tag{1.7}$$

and which gives the DSB the simple form

$$y(t) = \sum_{n=1}^{M} x(t-\tau_n) \tag{1.8}$$

The simple Delay-Sum Beamformer yields an improvement in SNR in the target direction, but its fixed choice of weights limits its ability to achieve optimum behavior for a particular acoustic scenario. For instance, if the weights are chosen correctly then the shape of the beampattern could be shifted to place one of its nulls directly over an interferer. Though this would be at the expense of weaker noise suppression elsewhere that fact might not matter if no other noise sources are present [5]. If the nature of the noise (its statistics in particular) is known *a priori* then optimal arrays can be designed ahead of time [6], but since audio scenes involving human talkers cannot be predicted and change rapidly an adaptive technique would be better. This is the motivation behind the study of adaptive array processing and is the focus of the next section.

1.3 Adaptive Beamforming

1.3.1 Frost's Algorithm

The Frost Algorithm [7] is the first attempt at finding a beamformer that applies weights to the sensor signals in an optimal sense. The setup for his system is shown in Figure 1.1 where it is assumed here and henceforth that the beamformer has already been steered (had each channel appropriately delayed) toward the target of interest. For the Frost Algorithm and from now on we recognize that our algorithms must be implemented on a digital computer, meaning that we reference all signals by an

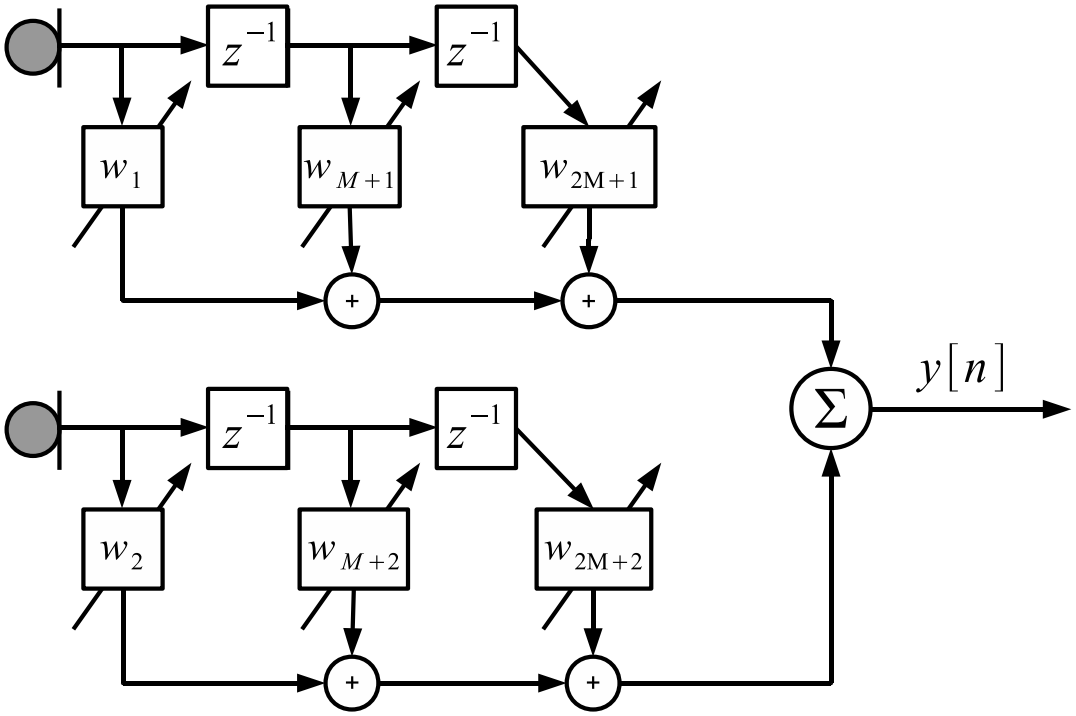

Figure 1.1: Frost's Beamformer

integer-valued index n and that we can store only so much of each received signal through a series of digital delay units.

The algorithm attempts to optimize the weighted sum of all input samples, expressed as

$$y[n] = \mathbf{W}^T \mathbf{X}[n] \tag{1.9}$$

where, in Frost's derivation, $\mathbf{X}[n]$ is a vector containing all samples of all channels currently stored in the beamformer and $\mathbf{W}$ is a vector of weights applied to each value in $\mathbf{X}[n]$. In general there are M sensors and O stored values for each sensor. The optimization attempts to minimize the expected output power of the beamformer, expressed as

$$\mathbb{E}\big(y^2[n]\big) = \mathbb{E}\big(\mathbf{W}^T \mathbf{X}[n] \mathbf{X}^T[n] W]\big) \tag{1.10}$$

$$= \mathbf{W}^T \mathbf{R}_{XX} \mathbf{W} \tag{1.11}$$

where $\mathbf{R}_{XX}$ is the correlation matrix of the input data and $\mathbb{E}$ is the expected value operator. The minimization is carried out under the constraint that sum of each

[1]Distortion will occur for a beampattern viewed as a function of receiving angles because D is a function of sines and cosines of θ and ϕ through $\boldsymbol{\alpha}$

column of weights in Figure 1.1 must equal some chosen number. If the vector of these numbers is expressed as

$$\mathcal{F} = [f_1 \quad f_2 \quad \dots \quad f_J] \tag{1.12}$$

the constraints take the form

$$\mathbf{C}^T\mathbf{W} = \mathcal{F} \tag{1.13}$$

where $\mathbf{C}$ is a matrix of ones and zeroes that selects the column weights in $\mathbf{W}$ appropriately. The vector $\mathcal{F}$ can be chosen as any vector of real numbers; one popular one that we'll use later is simply a digital delta function:

$$\mathcal{F} = [1\ 0\ 0\ 0 \dots] \tag{1.14}$$

What this choice would imply in Figure 1.1 is that the weights applied to the non-delayed elements w_1 and w_2 must sum to 1 and that the time-delayed elements w_{M+1} and w_{M+2} and w_{2M+1} and w_{2M+2} must each, in column-wise pairs as in the figure, sum to zero. This setup would mean that the target signal component arriving at the microphones (which would be completely identical at each sensor ideally) would pass through unchanged into $y[n]$, which is why this choice of constraints is called a *distortionless response*.

Now the optimization problem can be phrased as the constrained minimization problem

$$\underset{\mathbf{W}}{\text{minimize}} \quad \mathbf{W}^T\mathbf{R}_{XX}\mathbf{W} \tag{1.15}$$

subject to

$$\mathbf{C}^T\mathbf{W} = \mathcal{F} \tag{1.16}$$

This optimization is solved by the method of Lagrange Multipliers, which states that given an optimization problem of finding the extrema of some function f subject to the constraint $g = c$ for function g and constant c we can introduce a multiplier λ and find the extrema of the Lagrange function [8]

$$\Lambda = f + \lambda(g - c) \tag{1.17}$$

Here we compute the Lagrange function for the given target function and constraint as

$$H(\mathbf{W}) = \frac{1}{2}\mathbf{W}^T\mathbf{R}_{XX}\mathbf{W} + \boldsymbol{\lambda}^T(\mathbf{C}^T\mathbf{W} - \mathcal{F}) \tag{1.18}$$

The optimum is found by setting the gradient of this Lagrange function to zero, which can be shown to be

$$\nabla_W H(\mathbf{W}) = \mathbf{R}_{XX}\mathbf{W} + \mathbf{C}\boldsymbol{\lambda} = 0 \tag{1.19}$$

Hence the optimal weights are

$$\mathbf{W}_{opt} = -\mathbf{R}_{XX}^{-1}\mathbf{C}\boldsymbol{\lambda} \tag{1.20}$$

Now since the weights must still satisfy the constraint

$$\mathbf{C}^T\mathbf{W}_{opt} = \mathcal{F} = -\mathbf{C}^T\mathbf{R}_{XX}^{-1}\mathbf{C}\boldsymbol{\lambda} \tag{1.21}$$

the Lagrange multipliers can be explicitly solved for as

$$\boldsymbol{\lambda} = -\big(\mathbf{C}^T\mathbf{R}_{XX}^{-1}\mathbf{C}\big)^{-1}\mathcal{F} \tag{1.22}$$

which gives the optimal weight vector the form

$$\mathbf{W}_{opt} = \mathbf{R}_{XX}^{-1}\mathbf{C}\big(\mathbf{C}^T\mathbf{R}_{XX}^{-1}\mathbf{C}\big)^{-1}\mathcal{F} \tag{1.23}$$

The problem with this formulation, however, is that it assumes that the correlation matrix for the input, $\mathbf{R}_{XX}$, is stationary and known ahead of time. But since this isn't the case for an adaptive array, the weights need to be updated in a gradient descent fashion over time where, for every new sample of data, we modify the weights in the direction of the optimal weights:

$$\mathbf{W}[n+1] = \mathbf{W}[n] - \mu\nabla_W H(\mathbf{W}) \tag{1.24}$$

$$= \mathbf{W}[n] - \mu\big(\mathbf{R}_{XX}\mathbf{W} + \mathbf{C}\boldsymbol{\lambda}[n]\big) \tag{1.25}$$

where μ is an the adaptive step size parameter that controls how quickly the system adjusts at every iteration. We can solve for the Lagrange multipliers in this expression by substituting into the constraint equation

$$\mathcal{F} = \mathbf{C}^T\mathbf{W}[n+1] \tag{1.26}$$

$$= \mathbf{C}^T\mathbf{W}[n] - \mu\mathbf{C}^T\mathbf{R}_{XX}\mathbf{W}[n] - \mu\mathbf{C}^T\mathbf{C}\boldsymbol{\lambda}[n] \tag{1.27}$$

Solving this expression for $\boldsymbol{\lambda}[n]$ and plugging into the weight update equation yields

$$\mathbf{W}[n+1] = \mathbf{W}[n] - \mu\big(\mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\big)\mathbf{R}_{XX}\mathbf{W}[n]\ldots \tag{1.28}$$

$$+ \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\big(\mathcal{F} - \mathbf{C}^T\mathbf{W}[n]\big) \tag{1.29}$$

where $\mathbf{I}$ is the identity matrix. To simplify notation, define the following:

$$\mathbf{F} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathcal{F} \tag{1.30}$$

$$\mathbf{P} = \mathbf{I} - \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T \tag{1.31}$$

Furthermore, something still needs to be done about the unknown correlation matrix $\mathbf{R}_{XX}$. The quickest and easiest way to approximate this matrix is to simply take the outer product of the current value of the input vector with itself:

$$\mathbf{R}_{XX}[n] \approx \mathbf{X}[n]\mathbf{X}^T[n] \tag{1.32}$$

With these definitions, the final form of the Frost algorithm for updating towards the optimal filter taps is expressed as

$$\boxed{\mathbf{W}[n+1] = \mathbf{P}\big(\mathbf{W}[n] - \mu y[n]\mathbf{X}[n]\big) + \mathbf{F}} \tag{1.33}$$

1.3.2 The Generalized Sibelobe Canceller (Griffiths-Jim Beamformer)

The Generalized Sidelobe Canceller is a simplification of the Frost Algorithm presented by Griffiths and Jim some ten years after Frost's original paper was published [9]. Displayed in Figure 1.2, the structure consists of an upper branch often called the Fixed Beamformer (FBF) and a lower branch consisting of a Blocking Matrix (BM). (Note again that it is assumed that all input channels have already been appropriately steered toward the point of interest.)

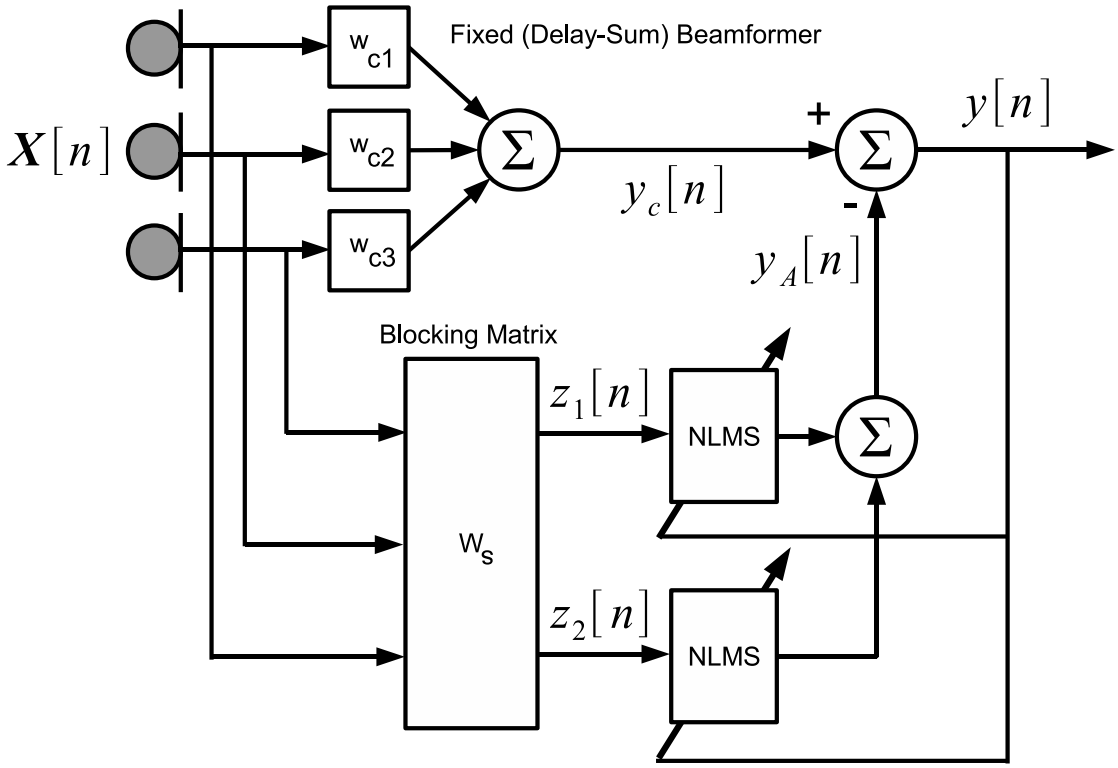

Figure 1.2: The Generalized Sidelobe Canceller

The upper branch is called a Fixed Beamformer because its behavior is constant over time. The constants w_c may be chosen as any nonzero values but are almost always chosen as simply $1/M$, yielding the traditional Delay and Sum beamformer:

$$y_c[n] = \frac{1}{M} \sum_{k=1}^{M} x_k[n] \tag{1.34}$$

(Remember that in current notation we assume that the sensors have already been target-aligned. In addition, we now adopt the more common practice of referencing

the input data and tap weights not as vectors but as matrices of size $O \times M$ where each column corresponds to data for an individual sensor.) The lower branch utilizes an unconstrained adaptive algorithm on a set of tracks that have passed through a Blocking Matrix (BM), consisting of some algorithm intended to eliminate the target signal from the incoming data in order to form a reference of the noise in the room. The particular BM used by Griffiths and Jim consists of simply taking pairwise differences of tracks, which would be visualized for the four-track instance as

$$\mathbf{W}_s = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \tag{1.35}$$

For this $\mathbf{W}_s$ the BM output tracks are computed as the matrix product of the blocking matrix and matrix of current input data.

$$\mathbf{Z}[n] = \mathbf{W}_s \mathbf{X}[n] \tag{1.36}$$

The overall beamformer output, $y[n]$, is computed as the DSB signal minus the sum of the adaptively-filtered BM tracks

$$y[n] = y_c[n] - \sum_{k=1}^{M-1} \mathbf{w}_k^T[n] \mathbf{z}_k[n] \tag{1.37}$$

where $\mathbf{w}_k[n]$ is the k^{th} column of the tap weight matrix $\mathbf{W}$ of length O and $\mathbf{z}_k[n]$ is the k^{th} Blocking Matrix output track, also of length O. The adaptive filters are each updated using the Normalized Least Mean Square (NLMS) algorithm with $y[n]$ as the reference signal

$$\mathbf{w}_k[n+1] = \mathbf{w}_k[n] + \mu y[n] \frac{\mathbf{z}_k[n]}{||\mathbf{z}_k[n]||^2} \tag{1.38}$$

A full explanation of how the GSC is derived from the Frost algorithm is beyond the scope of this work–the most important point is that it arises from ensuring that the sum of the weights for the DSB add to 1 and that the constraints for the Frost algorithm are chosen such that no distortion occurs for the target signal, which for an FIR filter means a digital delta function:

$$\mathcal{F}[n] = \delta[n] \tag{1.39}$$

1.4 Limitations of Current Models and Methods

The greatest problem observed thus far with the GSC is that, if the beamformer is incorrectly steered and doesn't point perfectly at its target, the target signal won't be completely eliminated after it has passed through the blocking matrix [5]. This problem will cause the adaptive filtering and subtracting stage to eliminate not just noise but some of the target waveform itself from the beamformer output and degrade performance. Corrections for steering errors have been tackled by some authors previously through the use of adaptive filters using the DSB output as reference [5],

though in a noisy environment the improvement will naturally be limited since even after the DSB stage the reference signal used will still be corrupted. Instead we propose a different statistical technique to compensate for incorrect steering where in Chapter 3 of this thesis we'll propose and evaluate a cross correlation technique that attempts to correct the beamformer lags.

In addition, the original formulations of the Frost and Griffiths-Jim algorithms were based on the general use of beamforming where the far-field assumption is often valid such as in radio astronomy or geology. But in this work, however, we're concerned with applying the GSC to an array implemented in an office that is at most several meters long and wide, meaning that the far field assumption is no longer valid. This change in the physics of the system will also cause leakage in the blocking matrix with the traditional Griffiths-Jim matrix because now the target signal is no longer received at each microphone with equal amplitude. Thus in Chapter 2 we study several amplitude adjustment models that attempt to overcome this problem.

And finally, much of the study of audio beamforming has been carried out with linear equispaced microphone arrays, due mostly to how arrays of other types of sensors have been constructed and how simple they are to understand mathematically. However, linear arrays are optimal only for a narrow frequency range that's dependent on the inter-microphone spacing and can be difficult to construct correctly, especially if surveillance is the intended application. Hence Chapter 4 will explore the effects of microphone geometry on beamforming performance and give guidelines on what makes for a good array.

1.5 Intelligibility and the SII Model

In human speech processing it's customary to evaluate the quality of a speech pattern in the presence of noise not in terms of a traditional SNR but a specially weighted scale called the Speech Intelligibility Index (SII) [10]. The index is calculated by running separate target and interference recordings through a bank of bandpass filters and multiplying the SNR for each frequency band by a weight based on subjective human tests. The calculation is expressed in notation as

$$SII = \sum_{n=1}^{N} A_n I_n \tag{1.40}$$

where N is the number of frequency bands under consideration ($N = 18$ here), A_n is the audibility of the n^{th} frequency band (essentially the SNR with some possible thresholding), and I_n is the n^{th} frequency band weight. The entire set of weights is referred to as the Band Importance function and is plotted in Figure 1.3.

The SII parameter ranges from 0 (completely unintelligible) to 1 (perfectly intelligible) and is computed over small windows of audio data, traditionally 20ms each, to yield a function of time. In this work the SII will be used to control the initial intelligibility of beamforming tests and provide a model for a simple FIR prefilter that can be applied to incoming audio data in order to ensure that the beamformer works solely on the frequency bands most important to human understanding of speech.

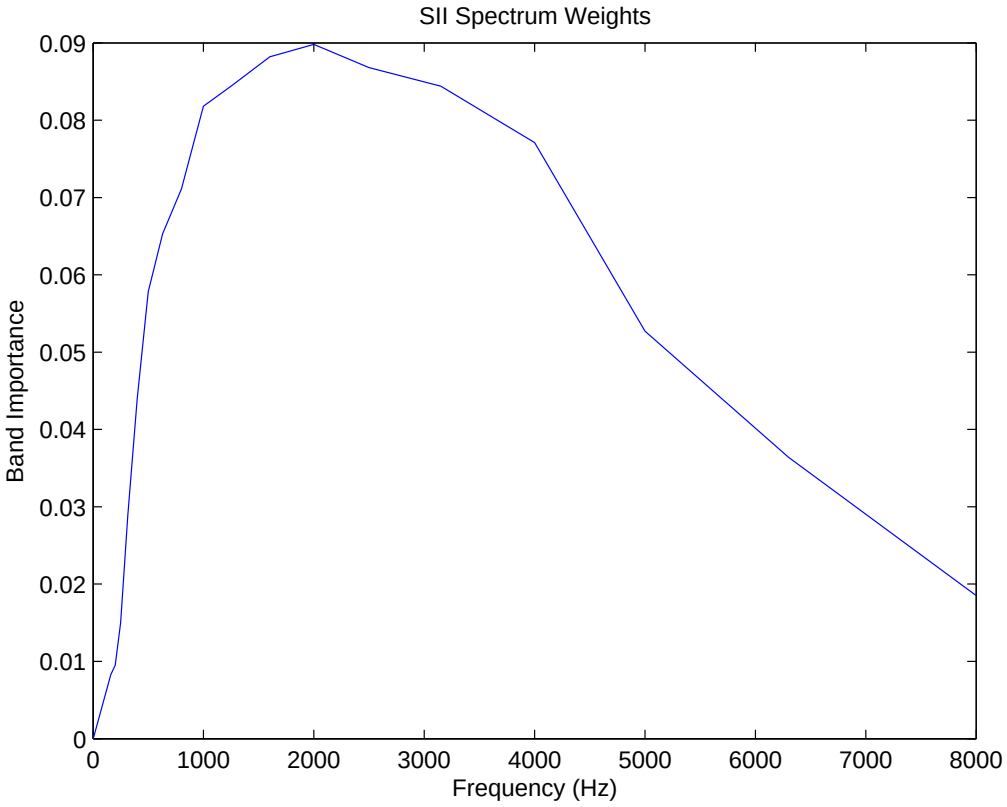

Figure 1.3: The SII Band Importance Spectrum

1.6 The Audio Data Archive

The experimental evaluations for this thesis are conducted using microphone array data collected over several months at the University of Kentucky's Center for Visualization and Virtual Environments. This data archive can be freely accessed over the World Wide Web [11] where full and up-to-date details on the archive can be found. In short, the data set consists of over a dozen different microphone array geometries in an aluminum cage several feet long and wide within a normal office environment. The 16-track recorded WAV files consist of both individual speakers at laser-measured coordinates and collections of human subjects talking to one another in order to simulate a cocktail party scenario, complete with clinking glasses and dishware. The human subjects include both males and females with varying ages and nationalities.

1.7 Organization of Thesis

Chapter 2 studies correcting the amplitude differences between signals entering the GSC Blocking Matrix to provide better target signal suppression by providing sev-

eral possible methods to enhance the pairwise subtraction and then evaluating each method over several sets of real audio data. Chapter 3 addresses correcting phase problems in the beamformer by using a windowed and thresholded cross correlation technique between pairs of tracks and evaluating whether this modification improves beamformer quality. Chapter 4 looks at the effects of microphone geometry through plots of multidimensional beampatterns and parameters for describing DSB beamfield quality. Chapter 5 sums up the research conducted for this work, and finally Appendix A provides a stability analysis for the GSC using z-transforms and a short computer verification.

Chapter 2

Statistical Amplitude Correction

2.1 Introduction

A sine wave at a particular frequency is completely determined by its amplitude and phase, and Fourier theory tells us that any recorded waveform can be viewed as a superposition of sine waves. Since one of the well-known weaknesses of the traditional GSC Blocking Matrix (BM) is that target signal leakage will degrade performance, from the Fourier standpoint one has two options to correct this problem: change the amplitudes in the BM or the phases. In Chapter 3 we address the use of cross correlation as a means of optimally estimating the phase difference between received target signal components, but here we propose and evaluate several techniques for dealing with the amplitude scaling that a sound wave experiences due to propagation through air to the microphones and distortion from the recording equipment. Two of the methods involve using models of the wave physics of the acoustic environment while one other proposes a statistical energy minimization technique in the frequency domain. In addition, we take advantage of how the audio data set for this thesis has been collected to show a method for simulating a perfect blocking matrix where no target signal is present whatsoever for comparison. The various methods are then compared using the correlation coefficient against the closest microphone track to the target speaker over many simulated cocktail parties.

2.2 Manipulating Track Order

Before going further, we present one very simple method of combating amplitude changes that will be utilized in all of our beamformers: switching track order based on distance.

The original GSC makes no distinctions about the order in which tracks should be processed–in fact, under its original farfield conditions the track order would be irrelevant since the target signal component would always be the same regardless of microphone-target distance. However, in the nearfield speaker distance will be a significant factor and will, at least in part, cause the target signal component to be received differently in all microphones. Hence microphones that are at similar

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

Figure 2.1: Example Griffiths-Jim Blocking Matrix for a Four-Channel Beamformer

distances to the target speaker will have more similar target components than mics that have more different distances. Expressed another way

$$A_k \propto d_k, \quad 1 \leq k < M \tag{2.1}$$

Since the goal here is to make the target signal component between pairs of tracks as similar as possible, an easy starting measure is to always sort the track orders and process in order from closest to furthest. Hence we force

$$d_k \leq d_{k+1} \forall k \tag{2.2}$$

This is a small change that, although it may or may not improve the beamform, has virtually zero computational cost as it only involves changing how we index into our BM tracks after sorting a handful of distances/delays. In addition, some of the models to be presented will work better if the mic distances are kept in order.

2.3 Models

As discussed in Chapter 1 a major problem with the GSC is leakage of the target signal through the Blocking Matrix (BM), causing the adaptive filters to erroneously eliminate target components from the overall beamformer output. This is due to the assumption in the algorithm's original derivation that the microphones receive identical target signals–a valid assumption for the beamformer's original radar application but not for the realm of nearfield audio beamforming. The original Griffiths-Jim blocking matrix makes this assumption especially conspicuous as it features the pair (1, -1) along the diagonal like in Figure 2.1 [9]. Several authors [5] [12] have addressed this issue through statistical means with adaptive filtering of blocking matrix channels using the Delay and Sum Beamformer (DSB) component as the reference signal. However, this method will still be prone to target signal leakage since the DSB will tend to achieve only moderate attenuation of at most a few decibels and hence a still-noisy signal will be used as the desired signal for the BM adaptive filters.

In order to attempt to minimize target signal leakage even further we propose and evaluate the following methods.

2.3.1 Spherical Wave Propagation in a Lossless Medium

The basic wave equation in spherical coordinates for an omnidirectional point sound source without boundaries is [13]

$$\frac{\partial p}{\partial r^2} + \frac{2}{r}\frac{\partial p}{\partial r} = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \tag{2.3}$$

where p is the sound pressure, r is the distance from the source, and c is the speed of sound. This differential equation has the solution [13]

$$p(r,t) = P_0 \frac{e^{j(\omega t - kr)}}{r} = P_0 \frac{e^{j(2\pi/\lambda)(ct-r)}}{r} \tag{2.4}$$

where P_0 is the amplitude at the source, $k = 2\pi/\lambda$, and $\omega = kc$. Solving the physics of acoustic wave propagation in this manner suggests a simple $1/r$ falloff in the amplitude of a sound independent of frequency.

One can use this simple inverse law to try to correct target signal amplitude scaling based purely on microphone-target distance by either 1. amplifying the signal at a further microphone or 2. attenuating the signal at a closer microphone. The wiser choice is the attenuation in order to avoid amplifying electronic noise. Such an algorithm could be visualized as in Figure 2.3 where one supposes that with Mic 1 at distance r_1 and Mic 2 at distance r_2 there exists a transfer function $H(r,\omega)$ that controls the shaping of the target signal $s[n]$ as it travels the distance r_1 to Mic 1 and that the same transfer function will operate over an additional distance $\Delta r_{1,2} = r_2 - r_1$ in cascade in order to transform the target signal received at Mic 1 to that received at Mic 2. The present model assumes that

$$H_{1/r}(r,\omega) = \frac{1}{r} \tag{2.5}$$

which implies the proportionality that for a signal with amplitude A_i at distance r_i and signal with amplitude at A_{i+1} at distance r_{i+1}

$$\frac{A_i}{A_{i+1}} = \frac{r_{i+1}}{r_i}, \quad 1 \leq i < M \tag{2.6}$$

In the blocking matrix we can assume that the further track has a relative amplitude of 1 so that the scaling for the closer track is

$$A_{i+1} = \frac{r_i}{r_{i+1}} \tag{2.7}$$

where, since we force the audio tracks to always be in order from closest to furthest from the target $r_i \leq r_{i+1} \; \forall \; i \Rightarrow A_i \leq 1 \; \forall \; i$, satisfying our desire to have the amplitude scaling always be an attenuation process. The resulting blocking matrix is displayed in Figure 2.2.

Advantages: Simple model, very low computational cost.
Disadvantages: Doesn't account for temperature, pressure, or humidity variations, room reverberations, equipment imperfections, or any other deviation from ideal.

2.3.2 Air as a Lossy Medium and the ISO Model

Although an inverse law is a good general model for the dissipation of sound energy as the wave propagates, the model assumes a lossless medium and therefore neglects

$$\begin{pmatrix} \frac{r_1}{r_2} & -1 & 0 & 0 \\ 0 & \frac{r_2}{r_3} & -1 & 0 \\ 0 & 0 & \frac{r_3}{r_4} & -1 \end{pmatrix}$$

Figure 2.2: Blocking Matrix for Spherical Lossless Model

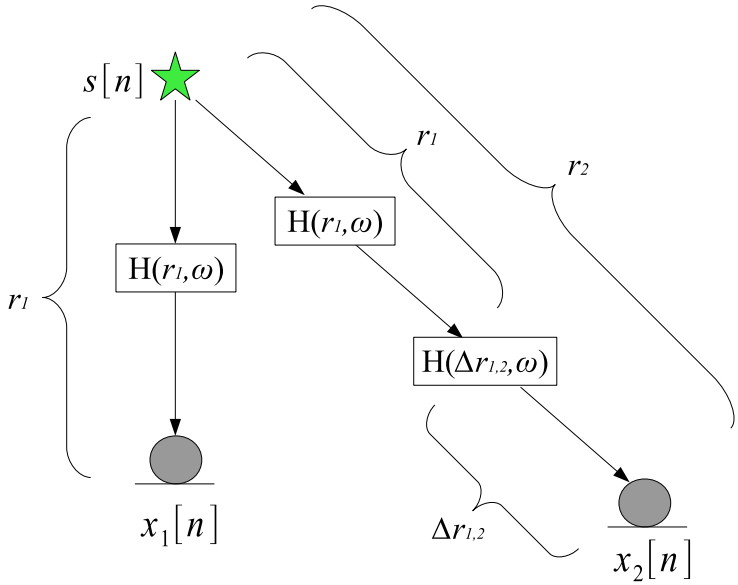

Figure 2.3: Sound Propagation Model as a Cascade of Filters

many of the fluid mechanical losses that a propagating acoustic wave experiences from the effects of viscosity, thermal conduction, and molecular thermal relaxation to name a few [14]. A full treatment of this subject is beyond the scope of this work but the subject has already been well-researched and the results codified in ISO 9613-1 (1993). To summarize, atmospheric sound attenuation is exponentially dependent on the distance the sound travels and a number dubbed the absorption coefficient, α_c (dB/m), which is a function of temperature, humidity, atmospheric pressure, and frequency. The result is a type of lowpass filter of form

$$H_{atm,dB}\left(r, \omega, T, P, h\right) = -r\alpha_c(\omega, T, P, h) \tag{2.8}$$

with r in meters, $\omega = 2\pi f$ the radial frequency with f in Hertz, T the temperature in Kelvin, P the atmospheric pressure in kPa, and h the relative humidity as a percentage. Computation of α_c is rather involved but can be quickly and easily implemented in software. Since α_c is frequency dependent we recognize that using the ISO model for a broadband signal amounts to a filtering operation. The frequency

$$\begin{pmatrix} H_{atm}\left(\Delta r_{1,2}, \omega, T, P, h\right) & -1 & 0 & 0 \\ 0 & H_{atm}\left(\Delta r_{2,3}, \omega, T, P, h\right) & -1 & 0 \\ 0 & 0 & H_{atm}\left(\Delta r_{3,4}, \omega, T, P, h\right) & -1 \end{pmatrix}$$

Figure 2.4: Blocking Matrix for ISO Sound Absorption Model in Frequency Domain

response of this filter can be generated by calculating several values of the absorption coefficient for $0 < f < f_s/2$ and then designing an FIR filter to match the response described by Eq 2.8. Thus the blocking matrix would be visualized as in Figure 2.4 where each closer track is filtered so that its target component matches that received at the farther microphone. This method will also result in a pure attenuation process, again ensuring that electronic noise is not unnecessarily amplified.

One potential drawback of this method, even if it's successful in target signal cancellation, is the fact that the filtering operation on the audio tracks will be applied to both the target and noise components of the tracks. This operation would thus shape the noise as it enters the MC stage of the beamformer and might present an unnatural change to the system.

Advantages: Very accurate model, uses easily-obtainable information to enhance beamforming.
Disadvantages: Increased computational cost for filtering, and if filter parameters change the filter design process must be repeated. Temperature, humidity, and atmospheric pressure must be measured. Doesn't account for room reverberations or electronic noise. May add distortion.

2.3.3 Statistical Blocking Matrix Energy Minimization

Though the ISO model takes several more environmental effects into account, by itself it also fails to consider noise within the electronic equipment, room reverberation, and speaker directivity. With so many factors affecting how the target sound is changed as it propagates to each of the microphones, we now propose a statistical method for amplitude correction that lumps all the corrupting effects together.

For a pair of real-valued random variables X and Y, it can be shown that if we wish to minimize the the squared error between between two variables using only a scalar multiplication on one, i.e.

$$(X - \alpha Y)^2 = e \tag{2.9}$$

then the constant α that will minimize the energy of the difference e is found as

$$\alpha = \frac{\mathbb{E}(XY)}{\mathbb{E}(Y^2)} \tag{2.10}$$

where $\mathbb{E}(\cdot)$ is the expected value operator. If we view the energy minimization problem in time domain where the audio data is always real we'd be done, but the distortions

occurring to the target sound has, at least in some part, a frequency dependence. So instead, let's generalize this result to the complex numbers so that a frequency-domain minimization can be carried out. In this case we express the energy as

$$(X - \alpha Y)(X - \alpha Y)^* = e \tag{2.11}$$

where * denotes complex conjugation. Applying the expected value yields

$$\mathbb{E}\Big((X - \alpha Y)(X - \alpha Y)^*\Big) = \mathbb{E}(e) \tag{2.12}$$

$$\mathbb{E}(XX^*) - \alpha\Big(\mathbb{E}(XY^*) + \mathbb{E}(X^*Y)\Big) + \alpha^2\mathbb{E}(YY^*) = \mathbb{E}(e) \tag{2.13}$$

The minimum energy is an extremum for α that can be found by taking the partial derivative with respect to α and solving.

$$\frac{\partial}{\partial\alpha}\left(\mathbb{E}(XX^*) - \alpha\Big(\mathbb{E}(XY^*) + \mathbb{E}(X^*Y)\Big) + \alpha^2\mathbb{E}(YY^*)\right) = \frac{\partial}{\partial\alpha}\Big(\mathbb{E}(e)\Big) \tag{2.14}$$

$$-\Big(\mathbb{E}(XY^*) + \mathbb{E}(X^*Y)\Big) + 2\alpha\mathbb{E}(YY^*) = 0 \tag{2.15}$$

$$\boxed{\alpha = \frac{\frac{1}{2}\big(\mathbb{E}(XY^*) + \mathbb{E}(X^*Y)\big)}{\mathbb{E}(YY^*)}} \tag{2.16}$$

This is one possible form of the scaling we wish to use. This expression can be rewritten in a more computationally-efficient way by noting that

$$\mathbb{E}(XY^*) + \mathbb{E}(X^*Y) = 2\mathrm{Re}\big(\mathbb{E}(XY^*)\big) \tag{2.17}$$

and

$$\mathbb{E}(YY^*) = \mathbb{E}(|Y|^2) \tag{2.18}$$

to get our final result where, since we wish to carry out the operation in frequency domain, X, Y, and α are all expressed as functions of angular frequency ω

$$\boxed{\alpha(\omega) = \frac{\mathrm{Re}\big(\mathbb{E}(X(\omega)Y^*(\omega))\big)}{\mathbb{E}(|Y(\omega)|^2)}} \tag{2.19}$$

(Remember again that we assume in our blocking matrix that X and Y have already been time-aligned to point the beamformer toward the desired focal point, hence no complex exponential phasing is shown.) Using this equation we can calculate a correction spectrum and apply it to the Fourier transforms of each pair of tracks entering the blocking matrix as

$$Z_k(\omega) = X_k(\omega) - \alpha_{k,k+1}(\omega)X_{k+1}(\omega) \tag{2.20}$$

Such a blocking matrix is visualized in Figure 2.5. This method will require continually estimating spectra for $X(\omega)$ and $Y(\omega)$ since these are audio tracks of human speech and hence nonstationary. However, voices are slowly-varying enough that if

$$\begin{pmatrix} 1 & -\alpha_{1,2}(\omega) & 0 & 0 \\ 0 & 1 & -\alpha_{2,3}(\omega) & 0 \\ 0 & 0 & 1 & -\alpha_{3,4}(\omega) \end{pmatrix}$$

Figure 2.5: Statistical Blocking Matrix in Frequency Domain.

we use an averaging technique of several windows on the order of 20ms a good estimate of the spectra can be generated. In addition, it's worthwhile to note that the spectrum computed in Eq 2.19 will be entirely real, meaning that it will target only the in-phase components between $X(\omega)$ and $Y(\omega)$ which should be the target signal components.

Now since we're forcing all tracks to be maintained in order from closest to furthest from the speaker, let's find a way to choose which of $X(\omega)$ and $Y(\omega)$ should be the closer track by analyzing how our statistical filtering will behave if we suppose a makeup of the signals $X(\omega)$ and $Y(\omega)$ of form

$$X(\omega) = H_1(\omega)S(\omega) + N_1(\omega) \tag{2.21}$$
$$Y(\omega) = H_2(\omega)S(\omega) + N_2(\omega) \tag{2.22}$$

where we let $S(\omega)$ be the target signal spectrum, $H_1(\omega)$ and $H_2(\omega)$ be the filters that shape the target signal components as they travel to the microphones whose signals are $X(\omega)$ and $Y(\omega)$, respectively, and $N_1(\omega)$ and $N_2(\omega)$ are lumped images of the noise within $X(\omega)$ and $Y(\omega)$, respectively. Now to get the target signal completely eliminated we would want

$$\alpha(\omega) = \frac{H_1(\omega)}{H_2(\omega)} \tag{2.23}$$

To see whether this will happen, we simply plug into Eq 2.16

$$\alpha(\omega) = \frac{\frac{1}{2}\big(\mathbb{E}(XY^*) + \mathbb{E}(X^*Y)\big)}{\mathbb{E}(YY^*)} \tag{2.24}$$

$$= \frac{\begin{array}{c}\mathbb{E}\Big(\big(H_1(\omega)S(\omega) + N_1(\omega)\big)\big(H_2(\omega)S(\omega) + N_2(\omega)\big)^*\Big) + \dots \\ \mathbb{E}\Big(\big(H_1(\omega)S(\omega) + N_1(\omega)\big)^*\big(H_2(\omega)S(\omega) + N_2(\omega)\big)\Big)\end{array}}{2\mathbb{E}\Big(\big(H_2(\omega)S(\omega) + N_2(\omega)\big)\big(H_2(\omega)S(\omega) + N_2(\omega)\big)^*\Big)} \tag{2.25}$$

To simplify this expression we note that the filters $H_1(\omega)$ and $H_2(\omega)$ are deterministic and can be taken outside of the expected value and assume that stochastic spectra $S(\omega)$, $N_1(\omega)$, and $N_2(\omega)$ are all uncorrelated such that an expected value of any of their products is zero. These considerations will lead to the simplification

$$\alpha(\omega) = \frac{\mathrm{Re}\big(H_1(\omega)H_2(\omega)\big)\mathbb{E}\big(|S(\omega)|^2\big)}{|H_2(\omega)|^2\mathbb{E}\big(|S(\omega)|^2\big) + \mathbb{E}\big(|N_2(\omega)|^2\big)} \tag{2.26}$$

This analysis shows that we should chose $Y(\omega)$ as the closer track since the closer track should tend to have a smaller noise component $N_2(\omega)$. This discussion also

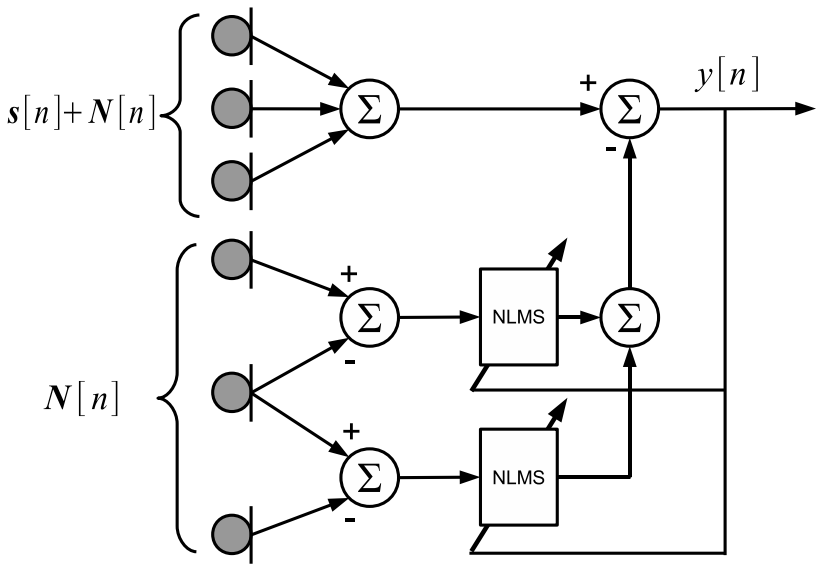

Figure 2.6: GSC Ideal Target Cancellation Simulation Signal Flow Diagram.

shows that, while we should chose $Y(\omega)$ as the closer mic between each pair of blocking matrix tracks, we also realize that the stronger the noise in the closer mic the greater the deviation in our correction spectrum from the ideal.

Advantages: Model tailored on the spot to an auditory scene by estimating current statistics, thus addressing all acoustic effects at once.
Disadvantages: Highest computational cost of the proposed models; correction spectrum becomes more distorted from ideal as the interference becomes stronger.

2.4 Simulating a Perfect Blocking Matrix

The data sets collected in the UK Vis Center's audio cage include separate recordings of individual speakers in a mostly quiet room and cocktail party recordings of several speakers. This separation gives us the convenient ability to piece scenarios together by simply adding together audio files. What we can do with this separation of target and noise is to feed them separately into the GSC as in Figure 2.6, where now we can truly observe a situation where the target signal never flows through the Blocking Matrix. This setup serves the two purposes of providing a benchmark for BM algorithm comparison as well as showing the ultimate limit on what any BM improvement can provide for overall GSC enhancement.

2.5 Experimental Evaluation

In order to test how well each model performs over many party-speaker positions and microphone array geometries, we chose an automated evaluation method using the Vis Center Audio Data archive described in Section 1.6. Combinations of a recording of a lone speaker and a recording of several interfering speakers were created so that the initial intelligibility [10] of the target speaker could be set to $.3 \pm .05$, a value considered a threshold for intelligibility. We choose a cross correlation method because:

1. An automated intelligibility test would require that the target and interference signals be completely separable, but the behavior of an adaptive system like the GSC is not linear–that is, the adaptation means that

$$GSC\Big(s[n] + v[n]\Big) \neq GSC\Big(s[n]\Big) + GSC\Big(v[n]\Big) \tag{2.27}$$

2. A traditional Mean Opinion Score (MOS) test would be very time consuming, especially if we want to gather a large amount of data.

We evaluated both the effectiveness of the blocking matrices and of the overall beamformers by finding the correlation coefficient with the closest microphone to the lone target speaker, the single best reference of the pure target signal. The correlation coefficient is computed for random vectors $\mathbf{x}$ and $\mathbf{y}$ as [15]

$$\rho_{xy}[m] = \frac{R_{xy}[m]}{||\mathbf{x}||||\mathbf{y}||} \qquad |\rho_{xy}| \leq 1 \tag{2.28}$$

where $R_{xy}[m]$ is the cross correlation between X and Y at lag m, defined as

$$R_{xy}[m] = \sum_{n=0}^{N-m-1} x[n+m]y[n] \tag{2.29}$$

The normalization by the product of norms for the correlation coefficient ensures that ρ_{xy} is bounded between -1 and 1. An effective blocking matrix should have a small correlation coefficient (eliminates the target well) while an effective overall beamformer should have a large correlation coefficient (recreates the target well). The relevant parameters to the beamformer are summarized in Table 2.1 and the correlation results displayed in Table 2.3 for the BM and Table 2.2 for the overall beamformers. Since there were three target speakers and three parties for each geometry the sample size is 9 for each beamformer situation (each of the three speakers gets placed individually into each of the three parties) and hence the sample size for each BM situation is 135 (nine speaker situations times fifteen BM tracks).

For the statistical energy minimization technique the length of the audio data segments we use becomes an issue due to the changing statistics of the environment. Here we use different segments of data for spectral estimation and the actual filtering– a shorter segment of data runs through the Blocking Matrix while a longer segment

Table 2.1: Parameters for Amplitude Correction Tests

Parameter	**Value**
Number of Microphone Channels	$M = 16$
Audio Sampling Rate	$f_s = 22.05$ kHz
NLMS Step Size	$\mu = .01$
NLMS Filter Order	$O = 32$
NLMS Forgetting Factor	$\beta = .95$
Audio Window Length	1024 samples
Spectral Estimation Data Length	4096 samples
Spectral Estimation Window	Tukey, $r = .25$
Closest Mic Initial Intelligibility	$.3 \pm .05$
ISO Filter Atmospheric Pressure	30 inHg
ISO Filter Temperature	$20°C$
ISO Filter Relative Humidity	40%

including and surrounding the shorter segment is used for power spectral density estimation associated with the processed segment. Since the FFT runs much faster when the number of points is a power of two, we chose the audio segment length to be 1024 (about 46ms of audio at $f_s = 22.05$ kHz) and the spectral estimation length to be 4096 samples (about 186 ms). For breaking apart the spectral estimation data a Tukey window was chosen with shape parameter $r = .25$.

2.6 Results and Discussion

The mean correlation coefficients for the overall GSC output with our different BM models are displayed in Table 2.2 and as a chart in Figure 2.7. Likewise, the mean correlation coefficients for the BM tracks using the different models are displayed in Table 2.3 and as a chart in Figure 2.8

Table 2.2: GSC Mean Correlation Coefficients, BM Amplitude Correction

	Microphone Geometry			
BM Method	**Linear**	**Rectangular**	**Perimeter**	**Random**
Traditional GSC	.564	.401	.349	.467
1/r Model	.565	.396	.347	.461
ISO Model	.580	.406	.351	.472
Statistical Model	.555	.376	.336	.456
Perfect BM	.631	.426	.376	.503

Table 2.3: BM Track Mean Correlation Coefficient for Various Arrays and Models

	Microphone Geometry			
BM Method	**Linear**	**Rectangular**	**Perimeter**	**Random**
Traditional GSC	.166	.105	.136	.138
1/r Model	.150	.106	.141	.140
ISO Model	.176	.137	.153	.157
Statistical Model	.215	.185	.175	.207
Perfect BM	.059	.059	.099	.062

For the Blocking Matrix we notice that, compared to the traditional Griffiths-Jim BM, the 1/r model performs slightly worse in all cases and the ISO filtering model slightly better. Our statistical filtering does a poor job of eliminating the correlation with the target signal while, as expected, the perfect BM does very well here. However, changes in BM performance have only a slight effect on overall beamformer performance, where a difference of as much as 15% in BM correlation improvement translates into only a 7% difference in the beamformer output correlation.

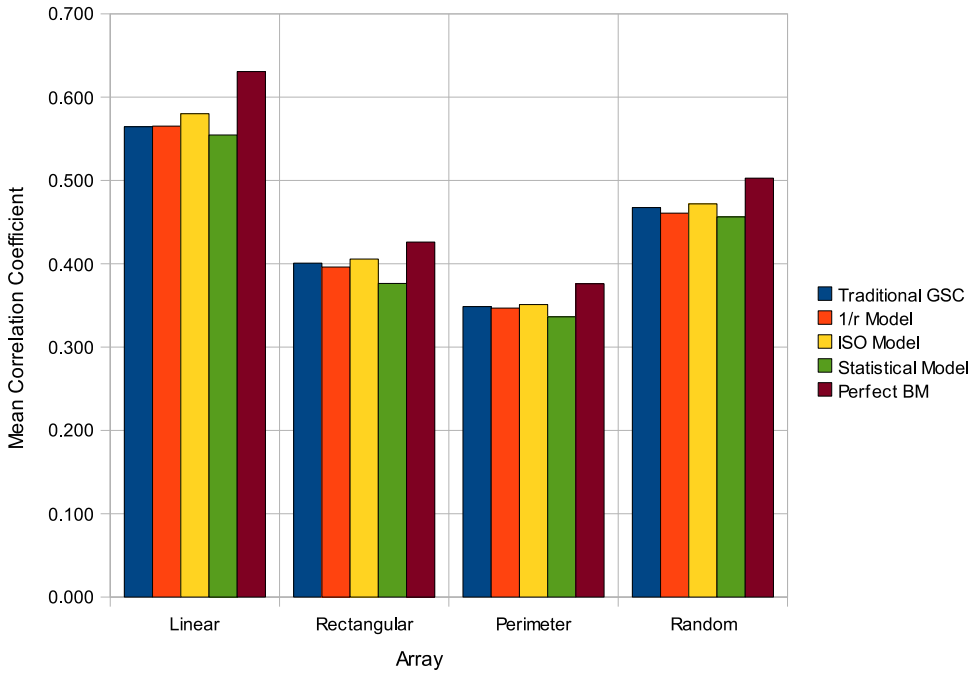

Figure 2.7: GSC Output Bar Chart for Data in Table 2.2

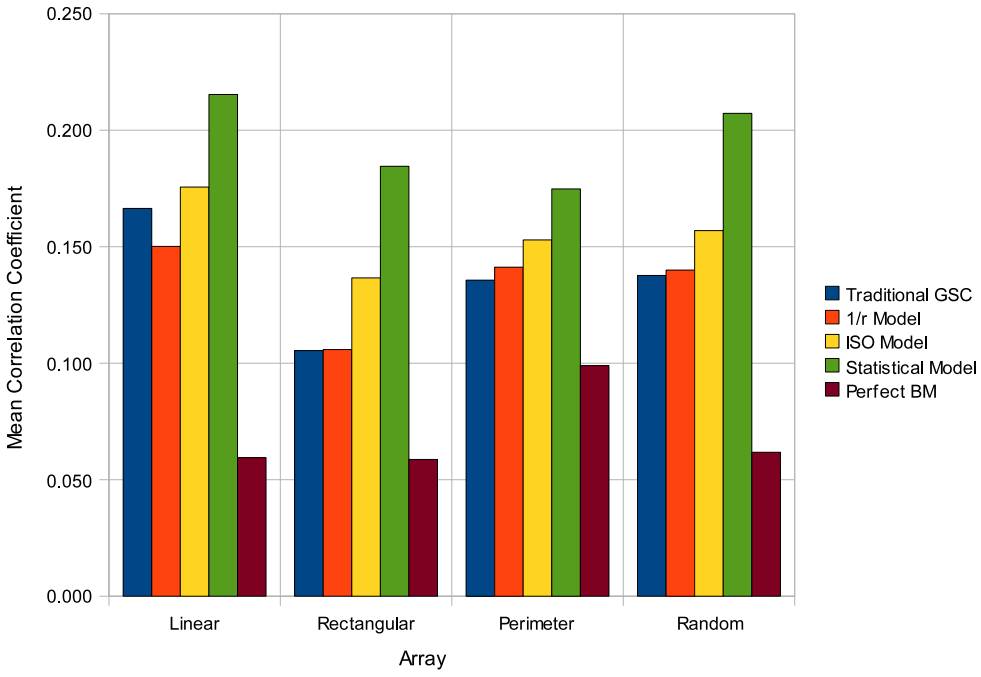

Figure 2.8: BM Bar Chart for Data in Table 2.3

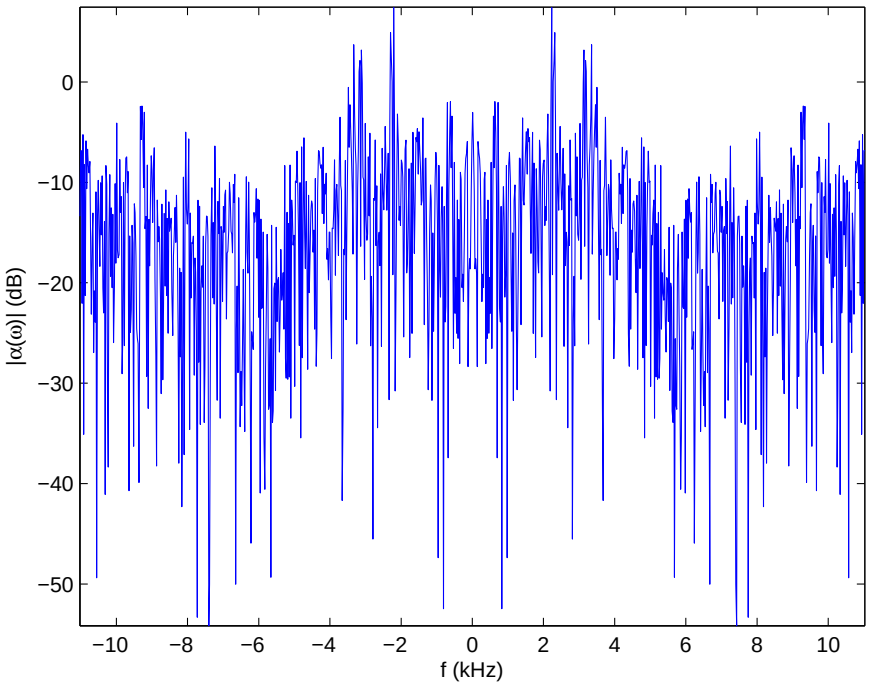

Figure 2.9: Sample Magnitude Spectrum for Statistical BM

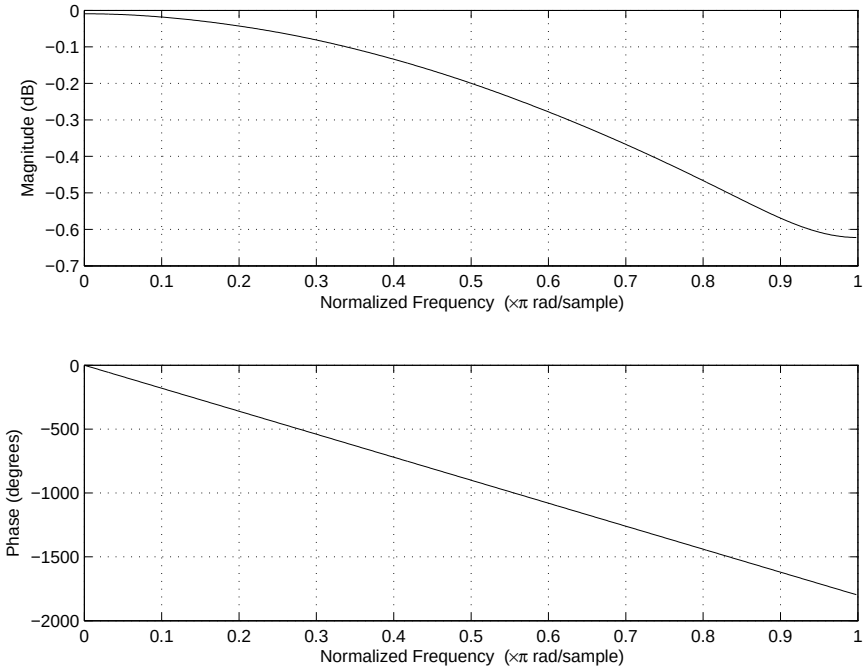

Figure 2.10: Magnitude and Phase Response for ISO Filter, $d = 3m$

To see why the statistical model seems to do so poorly, we present a sample of the computed correction spectrum in Figure 2.9. The example shows a very erratic magnitude response, varying over 50 dB. In contrast, an example of the ISO filter is presented in Figure 2.10 that shows a very smooth frequency response that spans less than one decibel. Since the ISO method works slightly better it would seem that such an extreme range of filtering as in the Statistical model is not appropriate. This erratic behavior may be due to the fact that, as previously noted, the statistical model performance is expected to deteriorate as the SNR worsens. And, since one would beamform only in a poor SNR scenario, these results suggest that the statistical method presented in this chapter may, therefore, not be useful at all.

Perhaps the most interesting result is the fact that the BM model used does not make as much of a difference as the microphone geometry in each experiment. All cases of the linear array, regardless of BM model, outperform all cases of the random array, with this pattern continuing in the same manner for the rectangular and perimeter arrays. Listening to some of the sample output tracks (available with the ETD) makes these statistical results readily apparent–the linear array output is significantly improved but the differences between the BM models is nearly impossible to hear save for the perfect BM, while with the perimeter array all models provide only a small improvement. This reliance on geometry is due to structure of the GSC, where the Delay-Sum portion of the beamformer is influenced only by the array geometry, and the results of this chapter indicate that the geometry is, in fact, more important to beamformer performance than any BM technique, even in the best case. In Chapter 4 we'll carry out an in-depth investigation into what geometries make for a good or bad microphone array.

2.6.1 Example WAV's Included with ETD

In order to immediately demonstrate the performance of each of the proposed algorithms the reader is invited to listen to some sample recordings included with this ETD the List of Files in the front matter of this thesis. Sample WAV's are provided for runs on the linear and perimeter arrays for the closest microphone to the target speaker alone, the closest microphone to the speaker in the constructed cocktail party, and overall GSC output tracks for each of the BM algorithms analyzed in this chapter.

The supplied WAV files should make it clear that, while the perfect blocking matrix does do slightly better, the different BM algorithms make very little difference in the overall beamformer output where the improvement is dominated by the array geometry (the improvement in intelligibility for the linear array is much greater than for the perimeter array in all cases).

2.7 Conclusion

In this chapter several methods for suppressing target signal leakage in the GSC BM were presented and their performance evaluated over several target-noise scenarios

for several different array geometries. Using the correlation coefficient against the closest microphone to the target speaker alone as reference, we determined that, in comparison to the traditional Griffiths-Jim blocking matrix, the $1/r$ and Statistical models performed slightly worse while the ISO model performed slightly better, both in terms of target signal leakage in the blocking matrix and overall beamformer performance. A theoretical perfect blocking matrix was also run and showed that even an ideal BM algorithm would be limited in improving the GSC overall.

Chapter 3

Automatic Steering Using Cross Correlation

3.1 Introduction

Errors in positional measurements for a microphone array are inevitable. Measured coordinates for each microphone will suffer whether measured with tape measure or laser and a target speaker's mouth will almost never remain in place or, in the case of surveillance, its position can obviously only be estimated. Chapter 2 addresses handling target signal leakage in the Blocking Matrix via amplitude adjustments but makes the assumption that the target position is exactly known, which is practically impossible. However, the cross-correlation is a well-known and highly-robust operation that can be used between microphone tracks on the fly to estimate the true speaker position. In this chapter we explain the Generalized Cross Correlation (GCC) procedure as presented in the literature along with a set of proposed improvements: application of bounds on how the much target can move for a windowed correlation search, and a threshold on how "certain" the calculations are as the correlation coefficient before any positional updates are made. We also present a simple multilateration technique that can allow for easy retracing from stored TDOA values to an exact Cartesian coordinate for a three-dimensional array. Finally, we fully evaluate how well the enhanced steering ability improves the overall GSC output.

3.2 The GCC and PHAT Weighting Function

We begin by quickly reviewing the original presentation of the GCC method for optimally estimating the TDOA of a wavefront over a pair of sensors [16] [17]. For a pair of microphones $n = 1, 2$, define the time delays that are required for a wave at some source position to reach each of the sensors as τ_1 and τ_2 and the TDOA as $\tau_{12} = \tau_2 - \tau_1$. The received signals at the microphones can be expressed in time

domain as

$$x_1(t) = s(t - \tau_1) * g_1(\mathbf{q}_s, t) + v_1(t) \tag{3.1}$$

$$x_2(t) = s(t - \tau_1 - \tau_{12}) * g_2(\mathbf{q}_s, t) + v_2(t) \tag{3.2}$$

$$\tag{3.3}$$

which expresses the mic signals as delayed versions of the target signal passed through a filter dependent on space and time combined with some noise. The GCC function is then defined as the cross correlation of the microphone signal spectra as

$$R_{12}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_{12}(\omega) X_1(\omega) X_2(\omega)^* e^{j\omega\tau} d\omega \tag{3.4}$$

where $\Psi_{12}(\omega)$ is a selectable weighting function chosen to make the optimal estimate easier to detect. This TDOA estimate is chosen as

$$\hat{\tau}_{12} = \underset{\tau \in D}{\operatorname{argmax}}\ R_{12}(\tau) \tag{3.5}$$

where D is a restricted range of possible delays. One possibility for the weighting function that has shown promise is the PHAT (Phase Transform)

$$\Psi_{12}(\omega) = \frac{1}{|X_1(\omega) X_2^*(\omega)|} \tag{3.6}$$

which has the effect of whitening the signal spectra. This is useful since the correlation operation shows the greatest peak for white noise which is, optimally, a delta function.

3.3 Proposed Improvements

The use of the GCC method for TDOA estimation in audio beamforming has received some attention in the literature previously but has been criticized for weak performance in multi-source and low SNR scenarios [16]. Thus in order to improve the GCC performance we propose the following modifications:

1. Enforce a criterion on how strong the correlation is between tracks before updating, rather than accepting the argmax every time. This should be especially helpful during periods of speaker silence since the argmax would be based purely on interference.

2. Begin with a seed value for the target speaker location as an explicit Cartesian point (s_x, s_y, s_z) and thereafter scan for correlation spikes over a small region around the previous focal point rather than the entire room. The smaller the region we examine, the less of a chance other erroneous correlation spikes will be detected.

3. Recent research has indicated that restraining the amount of whitening in the PHAT operation may improve localization capabilities [18], so utilize this variant of $\Psi_{12}(\omega)$ instead.

We now present our method in full notation.

3.3.1 Windowing of Data

First, the method of selecting chunks of audio data over time must be addressed for two reasons. For one, the length of the audio segments must be chosen short enough so that the assumption of short-time stationarity for a human voice is valid. In addition, if our algorithm varies the lags used for signal delay between windows then discontinuities will occur–if the lags shrink then data will be thrown out and if the lags grow then gaps will form. Thus we handle our data windowing as follows:

1. Carry out the algorithm on segments of audio 20ms in length, as is traditional in audio signal processing.

2. Process the windows with a 50% overlap at the start and combine them at the final output with a cosine-squared window. This will smooth-out discontinuities formed by changing lags since the cosine-squared window tapers to zero at its edges where the irregularities would occur.

3.3.2 Partial Whitening

Next, we choose to separate out the PHAT whitening and cross correlation operations so that the whitening is carried out first in frequency domain but the scan for the cross correlation peak is handled in time domain. Thus we begin by generating the whitened version of each of the microphone tracks as

$$\tilde{x}_k[n] = \mathcal{F}^{-1}\left\{\frac{X_k(\omega)}{|X_k(\omega)|^\beta}\right\} \qquad 0 < \beta < 1 \tag{3.7}$$

where we let the tilde denote the whitened version of $x_k[n]$, $X_k(\omega)$ is the spectrum of $x_k[n]$, and $\mathcal{F}^{-1}$ represents the inverse Fourier Transform. Note that we use the PHAT-β technique of partial whitening [18] by raising the magnitude spectrum in the denominator to a power less than one. In addition, the whitening spectrum is computed with a Hamming window applied in time domain before the FFT is carried out in order to cut down on ripples in the spectrum from the implied rectangular window.

3.3.3 Windowed Cross Correlation

The cross correlation between pairs of microphone tracks is then carried out on the whitened signals as

$$R^{(i)}_{k,k+1}[n] = \tilde{x}^{(i)}_k \star \tilde{x}^{(i)}_{k+1} \qquad 1 \le k < M \tag{3.8}$$

$$= \sum_{\xi=\tau^{(i)}_{k,k+1}-D}^{\xi=\tau^{(i)}_{k,k+1}+D} \tilde{x}^{(i)}_k[\xi]\tilde{x}^{(i)}_{k+1}[n+\xi] \tag{3.9}$$

where the superscript (i) indicates the number of the data window being processed (usually of length 20ms), ξ is the dummy variable of cross correlation, $\tau_{k,k+1}$ is the

TDOA between microphones k and $k+1$, and D is the bound on the number of cross correlation points we wish to evaluate around the current TDOA. If we take a maximum bound on the speed of a moving speaker as 10 m/s we can calculate the neighborhood as

$$D = 10\,\frac{f_s \Delta_{win}}{c} \tag{3.10}$$

with Δ_{win} the length of each segment of audio in seconds. For a 20ms window this sampling window corresponds to a bound of 20cm on the speaker's movement in any direction, and for a sampling rate $f_s = 22.05$ kHz this constitutes a limit of about 13 samples above and below the current TDOA. This bound on the cross correlation is much tighter than that used in the GCC methods in the past, where in effect an entire room several meters across could be searched.

The initial value for the lags is taken from a seed value for the target speaker position from the Euclidean distance between the supplied speaker position and the microphone coordinates that the algorithm refines every Δ_{win} seconds thereafter. Hence

$$\tau_k^{(1)} = \frac{f_s}{c}\sqrt{(x_k - s_x)^2 + (y_k - s_y)^2 - (z_k - s_z)^2} \qquad 1 \leq k < M \tag{3.11}$$

where each microphone in the array is located at spatial coordinate (x_k, y_k, z_k).

3.3.4 Correlation Coefficient Threshold

Our update thresholding algorithm uses the correlation coefficient, which can be expressed in terms of the above cross correlation as [15]

$$\rho_{k,k+1}[n] = \frac{R_{k,k+1}[n]}{||\mathbf{x}_k||||\mathbf{x}_{k+1}||} \qquad |\rho_{k,k+1}| \leq 1 \forall n \tag{3.12}$$

where the normalization by the norms of the windows of the mic signals has the effect that the correlation coefficient will always range from ± 1 (perfectly correlated) to 0 (completely uncorrelated). We make use of the correlation coefficient to define our restrained TDOA update as

$$\tau_{k,k+1}^{(i+1)} = \begin{cases} \underset{n}{\text{argmax}}\ \rho_{k,k+1}^{(i)}[n] & \text{if } \underset{n}{\text{argmax}}\ \rho_{k,k+1}^{(i)}[n] > \rho_{thresh} \\ \\ \hat{\tau}_{k,k+1}^{(i)} & \text{otherwise} \end{cases} \tag{3.13}$$

where ρ_{thresh} is a chosen threshold between 0 and 1 that has the effect of requiring a defined amount of correlation between the whitened signals within the search window before a TDOA update can take place.

3.4 Multilateration

The automatic tracking provided by the correlative update for the beamformer lags provides a method of sound source tracking that, through a bit of algebraic manipulation, can yield an estimate of the Cartesian (x, y, z) position of the target, since the

number of lags required for a sound to reach a microphone is directly proportional to the Euclidean distance. In $\mathbb{R}^3$ any combination of three distances would uniquely determine the position of the target, but since in general $M > 3$ for a microphone array we are presented with an overdetermined system since more information is provided than there are parameters to be determined. However, this extra information over the array allows us to make a calculation over the entire array that minimizes the error over all sets of lags in the least-squares sense. This multilateration algorithm provides a very efficient method for sound source location and is derived as follows:

Suppose that the positions of the M microphones in an array are precisely known in $\mathbb{R}^3$, denoted as $(x_1, y_1, z_1), (x_2, y_2, z_2), ..., (x_M, y_M, z_M)$, and that the lags for a beamform for speed of sound c and sampling rate f_s are also known as $\tau_{1...M}$. We wish to solve for the position of the target (s_x, s_y, s_z). Firstly, the distances from each microphone to the target follow directly from the lags as

$$\tau_i = d_i \frac{f_s}{c} \quad 1 \leq i \leq M \tag{3.14}$$

Each of these distances is related the positions of the i^{th} microphone to the source by the formula for Euclidean distance

$$d_i = \sqrt{(x_i - s_x)^2 + (y_i - s_y)^2 + (z_i - s_z)^2} \quad 1 \leq i \leq M \tag{3.15}$$

or, by squaring both sides

$$d_i^2 = (x_i - s_x)^2 + (y_i - s_y)^2 + (z_i - s_z)^2 \quad 1 \leq i \leq M \tag{3.16}$$

Now what we would like to do is formulate a system of equations using these distance relationships that would allow us to solve for (s_x, s_y, s_z), but in the present form the squared terms for the source position are problematic if we wish to take a linear algebra route. However, those terms can be eliminated by expanding and taking differences of equations. If we expand Eq (3.16) and write the terms for both the i and $i + 1$ case we have

$$x_i^2 - 2x_i s_x + s_x^2 + y_i^2 - 2y_i s_y + s_y^2 + z_i^2 - 2z_i s_x + s_z^2 = d_i^2 \tag{3.17}$$

$$x_{i+1}^2 - 2x_{i+1} s_x + s_x^2 + y_{i+1}^2 - 2y_{i+1} s_y + s_y^2 + z_{i+1}^2 - 2z_{i+1} s_x + s_z^2 = d_{i+1}^2 \tag{3.18}$$

If we subtract the second line from the first, the squared terms for the source position disappear:

$$x_i^2 - x_{i+1}^2 - 2s_x(x_i - x_{i+1}) + y_i^2 - y_{i+1}^2 - 2s_y(y_i - y_{i+1}) + z_i^2 - z_{i+1}^2 + 2s_z(z_i - z_{i+1}) = d_i^2 - d_{i+1}^2 \tag{3.19}$$

Now we can rearrange this equation so that only terms involving the target position are on one side as

$$2s_x(x_{i+1} - x_i) + 2s_y(y_{i+1} - y_i) + 2s_z(z_{i+1} - z_i) = ... \tag{3.20}$$

$$d_i^2 - d_{i+1}^2 + x_{i+1}^2 - x_i^2 + y_{i+1}^2 - y_i^2 + z_{i+1}^2 - z_i^2 \tag{3.21}$$

Notice that all terms on the righthand side are known ahead of time. For the $M-1$ differences in distance that can be calculated we can write out Eq (3.20) $M-1$ times. In matrix form this would be

$$2\begin{pmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_2 & y_3 - y_2 & z_3 - z_2 \\ \vdots & \vdots & \vdots \\ x_M - x_{M-1} & y_M - y_{M-1} & z_M - z_{M-1} \end{pmatrix} \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} =$$

$$\begin{pmatrix} d_1^2 - d_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2 + z_2^2 - z_1^2 \\ d_2^2 - d_3^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2 + z_3^2 - z_2^2 \\ \vdots \\ d_{M-1}^2 - d_M^2 + x_M^2 - x_{M-1}^2 + y_M^2 - y_{M-1}^2 + z_M^2 - z_{M-1}^2 \end{pmatrix} \tag{3.22}$$

where the matrix dimensions are $(M-1 \times 3)$, (3×1), and $(M-1 \times 1)$, respectively. Now we can use the simple fact from linear algebra that, for an overdetermined system of form $\mathbf{Ax} = \mathbf{b}$, the least squares solution of the system is found as

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{3.23}$$

If we let $\mathbf{A}$ be the first matrix of Eq (3.22), $\mathbf{x}$ be the middle vector, and $\mathbf{b}$ be the final vector, then the position vector of the target can be solved for using Eq (3.23).

Though this algorithm requires a seed value for target position since it uses the lags from the modified GSC, its automatic tracking ability is a very attractive feature versus sound source location (SSL) schemes that essentially require beamforming over many points through some volume of space per every timeframe of audio. Correlation and multilateration, however, are fast operations that need to be run only once per frame of audio data and thus have the potential for great computational savings.

One interesting limitation of this algorithm is that its ability to find a target position can be limited by the geometry of the array for the special cases of planar and linear microphone arrays. For the case of a planar array the z-coordinate of all microphones will be the same, thus forcing the rightmost column of the first matrix in Eq (3.22) to be zero. But if we attempt to solve using (3.23) the inverse of $\mathbf{A}$ will not exist since $\mathbf{A}$ will be rank-deficient (rank at most 2 for an $M-1 \times 3$ matrix).

3.5 Experimental Evaluation

3.5.1 GSC Performance with Automatic Steering

To evaluate how the cross correlation updates for the array steering lags affect GSC performance, we repeated the correlation comparison technique used for evaluation in Chapter 2 where the speaker intelligibility was set to around .3 and the correlation coefficient was found between the beamformer output and the closest mic to the

Table 3.1: GSC Mean Correlation Coefficients, Automatic Steering

	Microphone Geometry			
ρ_{thresh}	**Linear**	**Rectangular**	**Perimeter**	**Random**
.1	.494	.280	.324	.399
.2	.526	.298	.329	.403
.3	.527	.288	.332	.410
.4	.513	.339	.341	.409
.5	.523	.376	.347	.428
.6	.531	.389	.347	.442
.7	.547	.398	.347	.458
.8	.552	.402	.347	.459
.9	.561	.402	.347	.463

Table 3.2: BM Mean Correlation Coefficients, Automatic Steering

	Microphone Geometry			
ρ_{thresh}	**Linear**	**Rectangular**	**Perimeter**	**Random**
.1	.210	.131	.174	.173
.2	.210	.131	.169	.169
.3	.208	.130	.169	.168
.4	.204	.128	.167	.165
.5	.200	.127	.166	.164
.6	.198	.126	.166	.164
.7	.197	.126	.166	.164
.8	.197	.125	.166	.164
.9	.196	.125	.166	.163

target speaker. (Refer back to Table 2.1 for system parameters). Since the choice of amplitude correction method made little difference in Chapter 2 the simplest approach, the traditional Griffiths-Jim pairwise subtraction, is used. The parameter ρ_{thresh} was chosen to vary from .1 to .9 and again the correlation between the target signal and both the BM tracks and overall GSC output was measured. The results are displayed in Tables 3.1 and 3.2 and visualized in Figures 3.1 and 3.2 for the GSC output and BM tracks, respectively.

3.5.2 Multilateration Versus SRP

The multilateration technique presented in this work requires a fully three-dimensional array in order to find a least-squares coordinate in $\mathbb{R}^3$. Of the arrays in the UK Vis Center Data Archive, three fit into this category (all others are either 2D or linear).

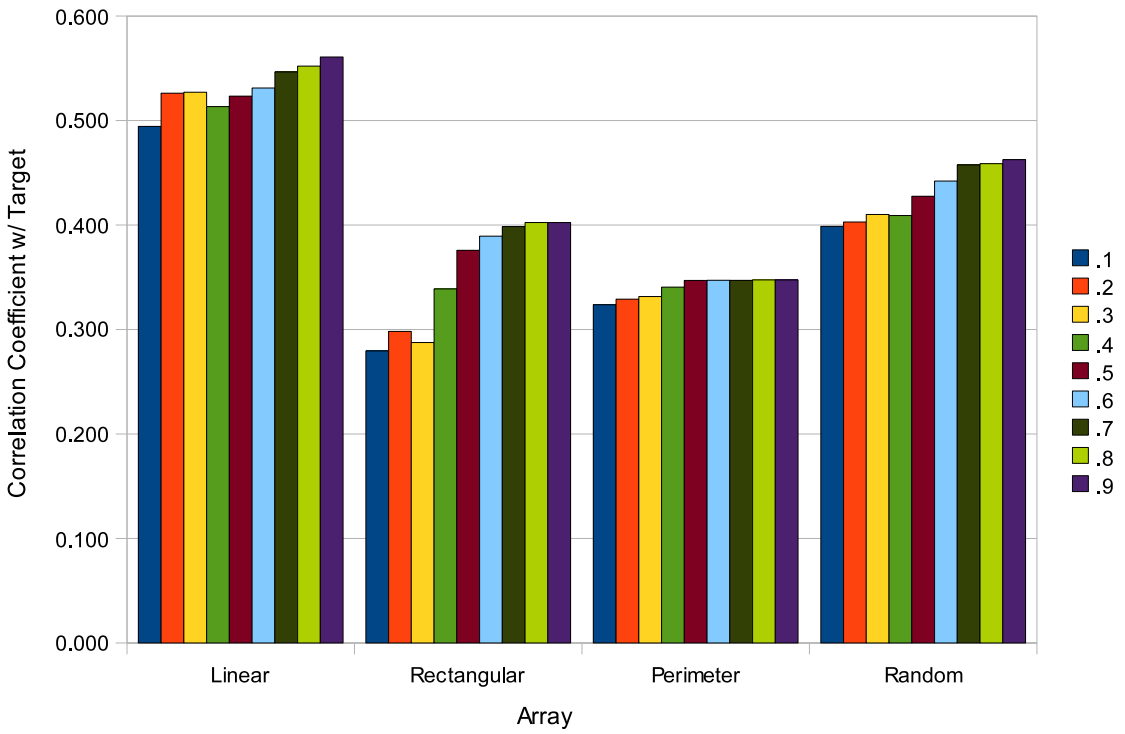

Figure 3.1: Bar Chart of GSC Output Track Correlations w/ Target

The data archive includes target speaker positions calculated by the SRP-PHAT sound source location technique [19]. For each of these arrays, we chose to run multilateration on the lags calculated by the thresholded cross correlation for $\rho_{thresh} = .1$, to .9 by increments of .1 and then calculate the mean Euclidean distance between the calculated points as

$$e = \frac{1}{N_{pts}} \sum_{i=1}^{N_{pts}} \sqrt{(x_{i,M} - x_{i,SSL})^2 + (y_{i,M} - y_{i,SSL})^2 + (z_{i,M} - z_{i,SSL})^2} \tag{3.24}$$

where N_{pts} is the number of points that SSL calculated over the entire audio track. N_{pts} may not and usually doesn't equal the number of 20 ms windows for the entire track since the SSL technique won't always detect a target speaker, especially when the talker is silent. We find this mean distance and the beamformer output correlation with the closest mic track to the target speaker alone as we again vary the correlation threshold from .1 to .9. The results are displayed in Tables 3.3 and 3.4 for the output correlations and errors, respectively, and visualized in Figures 3.3 and 3.4.

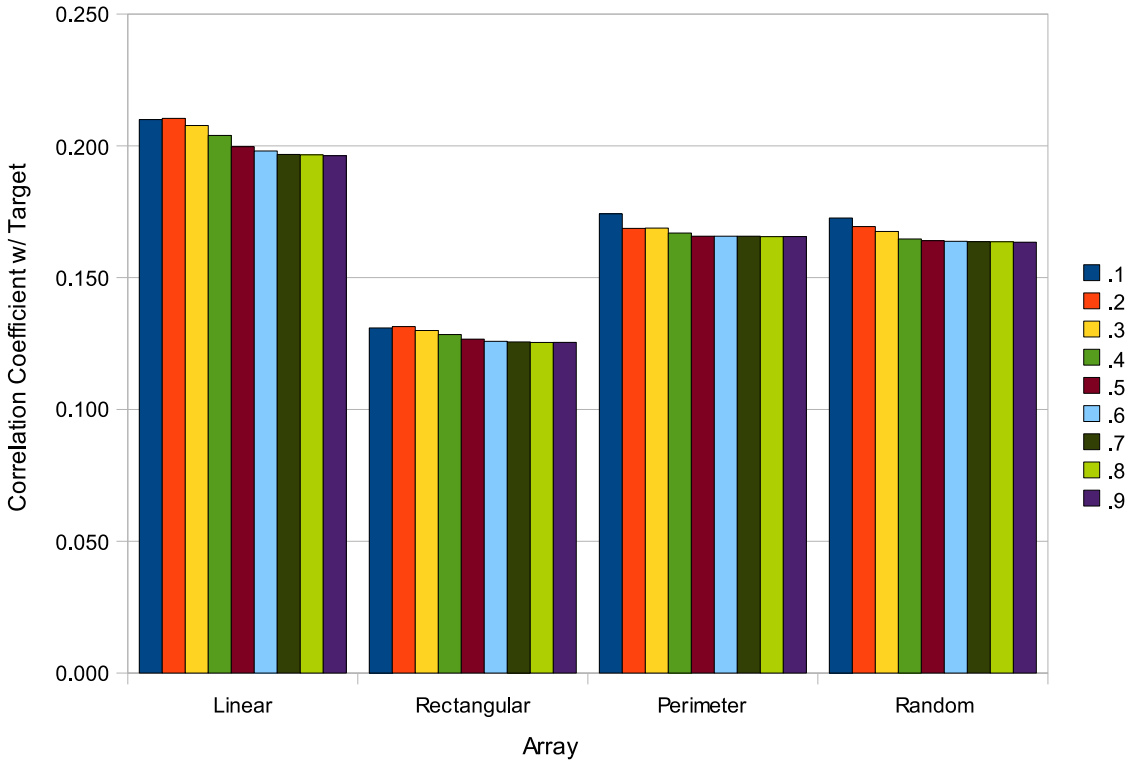

Figure 3.2: Bar Chart of BM Output Track Correlations w/ Target

Table 3.3: Beamformer Output Correlations for Various Thresholds

	Microphone Geometry		
ρ_{thresh}	**Endfire Cluster**	**Pairwise Even 3D**	**Spread Cluster**
.10	.2190	.2420	.238
.20	.2530	.2550	.280
.30	.2560	.2960	.273
.40	.2720	.3320	.280
.50	.2720	.3620	.294
.60	.2800	.3660	.302
.70	.2780	.3620	.307
.80	.2770	.3790	.320
.90	.2760	.3820	.318

Table 3.4: Mean Multilateration Errors vs SSL for Various Thresholds

	Microphone Geometry		
ρ_{thresh}	**Endfire Cluster**	**Pairwise Even 3D**	**Spread Cluster**
.1	3.625	1.444	0.285
.2	5.654	1.587	1.290
.3	5.578	1.599	1.263
.4	5.651	1.591	1.344
.5	5.888	1.571	1.367
.6	6.132	1.571	1.364
.7	6.173	1.573	1.366
.8	6.174	1.570	1.371
.9	6.192	1.570	1.368

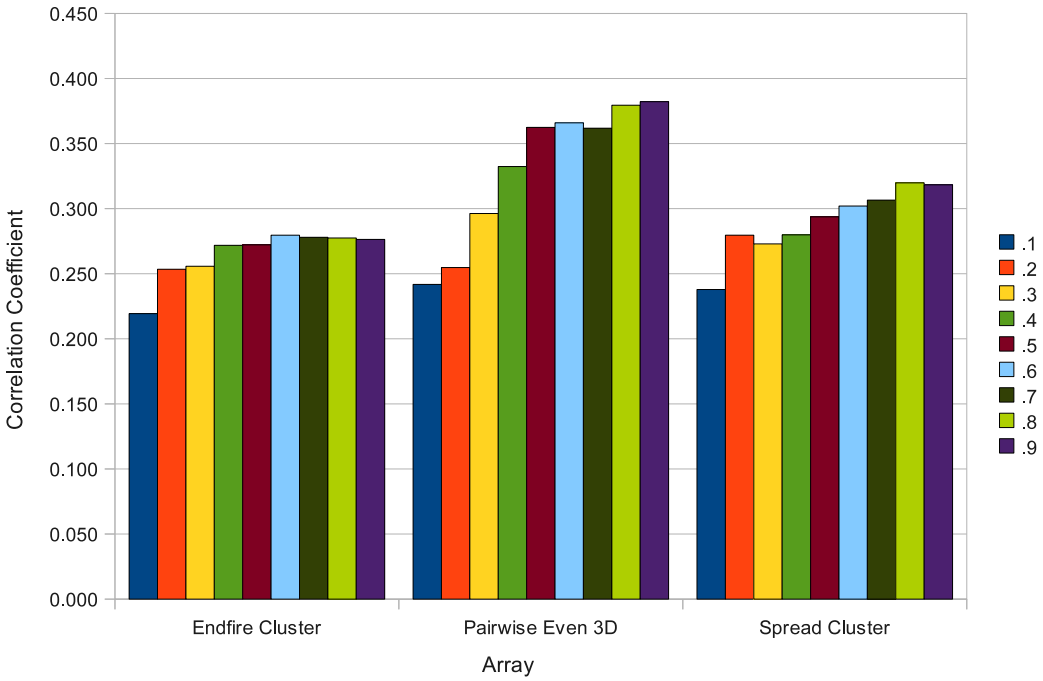

Figure 3.3: Bar Chart of Correlations from Table 3.3

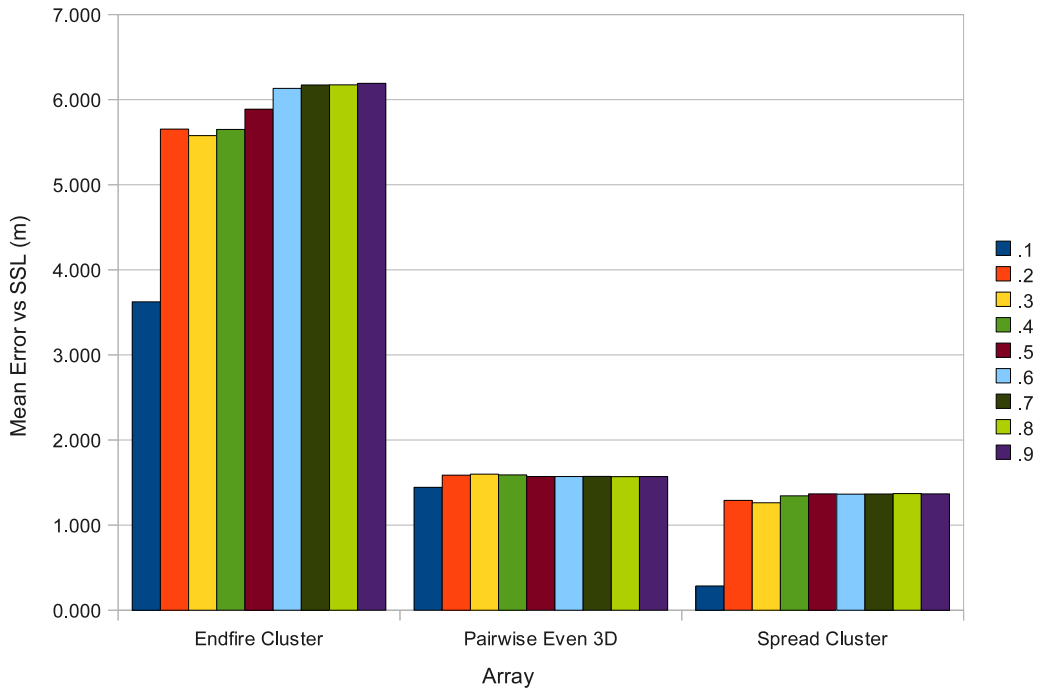

Figure 3.4: Bar Chart of Mean Errors vs SSL from Table 3.4

3.6 Results and Discussion

Since the target speaker has been held stationary for all recordings in the data archive, we expect that the only improvements for target steering would come from very small adjustments accounting for the tiny movements of a person's body as he speaks. Given this fact, we would expect a very high correlation coefficient threshold to be appropriate, and as Tables 3.1 and 3.2 show this is certainly the case. In fact, the results for the four arrays as used in Chapter 2 seem to suggest that the only good scenario, given that it's known the target is still, is to use no updating at all. This fact again shows the difficulty of using statistical methods in an inherently poor SNR situation.

In order to further investigate the correlation scheme's performance we examine the results of the multilateration tests, which will allow us to see a fully 3D rendering of where the beamformer "thinks" the target is at some instant. The results are displayed in Tables 3.3 and 3.4 and visualized in Figures 3.3 and 3.4.

What's interesting to see here is that the mean error between multilaterion over the adjusted lags and SSL doesn't change a great deal as the threshold for updating the alignment lags increases. To see why this is so, we take a look at some sample plots for the Endfire Cluster array of both the multilateration versus SSL points and the raw lags in the beamformer for thresholds of .1, .5, and .9. The points are plotted in Figures 3.5, 3.6, and 3.7 and the lags in Figures 3.8, 3.9, and 3.10.

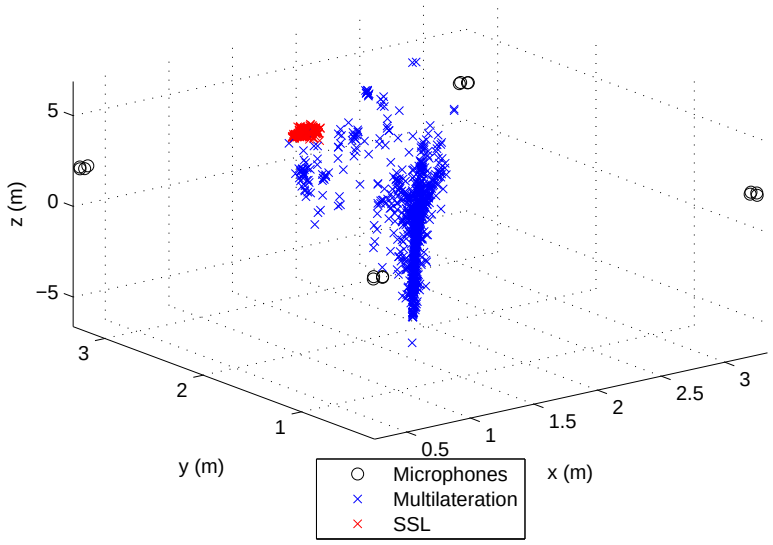

Figure 3.5: Multilateration and SSL Target Positions, $\rho_{thresh} = .1$

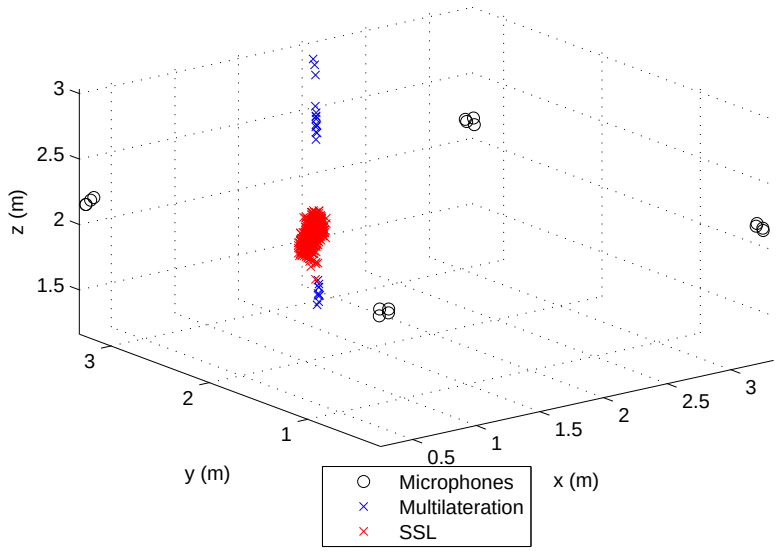

Figure 3.6: Multilateration and SSL Target Positions, $\rho_{thresh} = .5$

The positional plots show that the thresholding is working to some degree–the

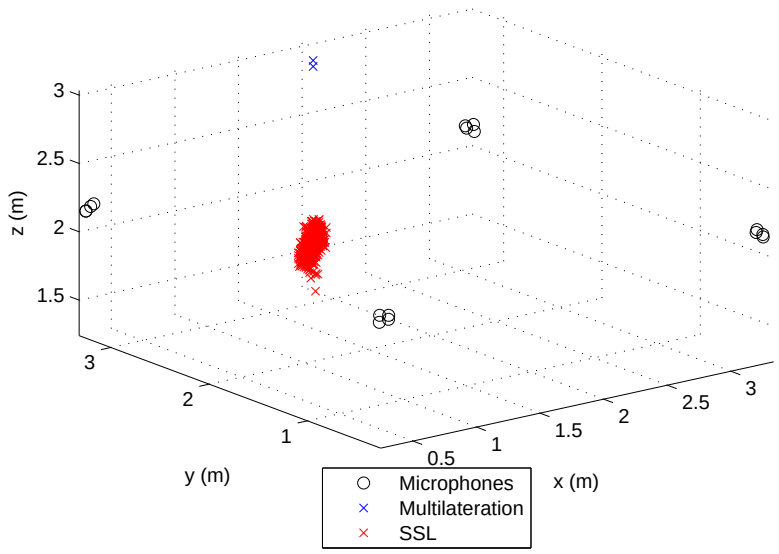

Figure 3.7: Multilateration and SSL Target Positions, $\rho_{thresh} = .9$

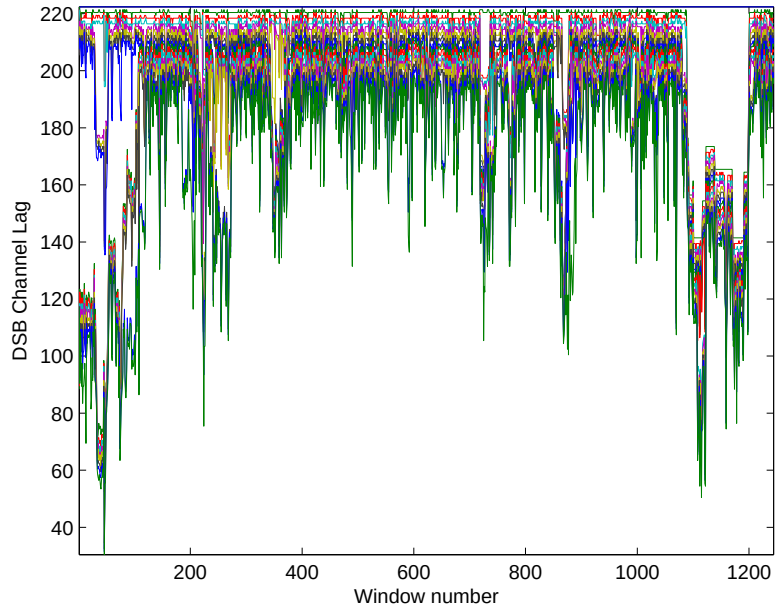

Figure 3.8: Multilateration and SSL Target Positions, $\rho_{thresh} = .1$

higher that threshold, the less often the focal point of the array will shift. For a low

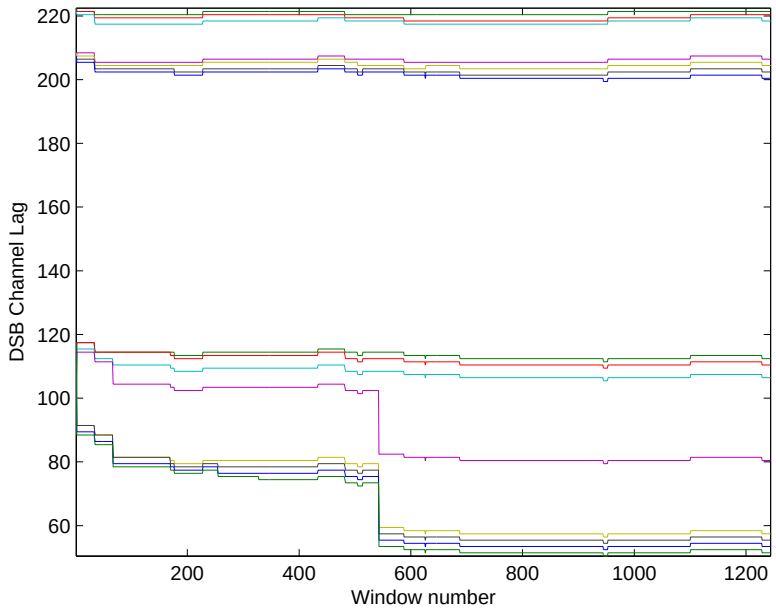

Figure 3.9: Multilateration and SSL Target Positions, $\rho_{thresh} = .5$

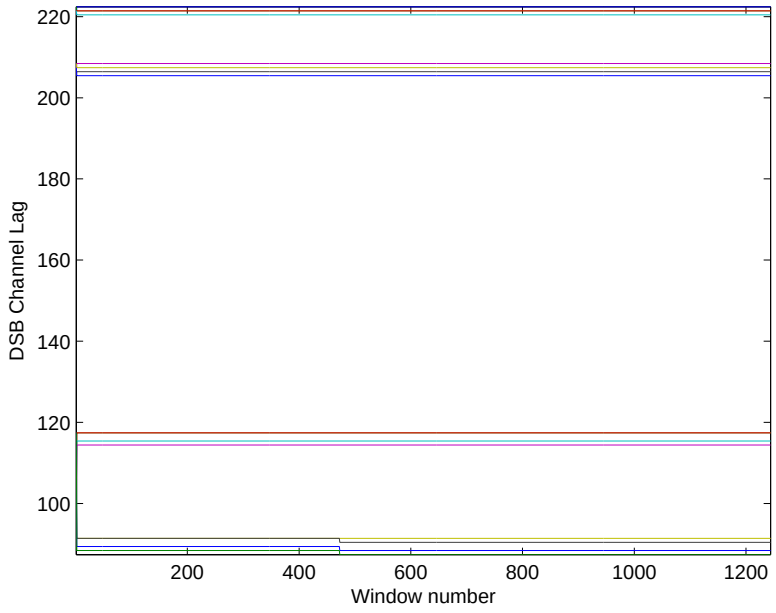

Figure 3.10: Multilateration and SSL Target Positions, $\rho_{thresh} = .9$

threshold like .1 the focal point moves very often and rather erratically, even moving

beyond the bounds of the room, while for a high threshold like .9 there are very few adjustments. At the same time, we notice that the low threshold plot indicates an ability to return to the correct focal point even after a large misadjustment since there are many points clustered around the SSL points as well as far away. On the other hand, the small number of points for the high threshold plot indicate that while a bad adjustment may be rare, undoing a bad adjustment is also as rare. These facts seem to indicate a potential tradeoff between low and high correlation thresholds: a low threshold is more likely to go off track but can recover more easily, while a high threshold is less likely to readjust incorrectly but has a far more difficult time recovering if it does.

The most revealing result of the Multilateration plots is that, despite our limitation that the target can move no more than 20cm in a 20ms time frame, we notice in Figure 3.7 that the least squares retraced focal point can jump by as much as a meter over a single frame. This fact suggests that enforcing a much smaller window on the correlation may help, perhaps because the 20cm window is enforced on each *pair* of tracks and not the entire array, meaning that in the worst case the distance limit compounds.

Finally, it's again worth mentioning that all audio data analyzed from the Vis Center archive involves stationary targets and interferers, which may give an unfair bias towards never adjusting the focal point. An interesting piece of future work would be be an investigation of how the presented tracking scheme would behave for a moving target speaker and how it would perform against SSL, especially when the target speaker has longer periods of silence as he moves. This would likely require an enlarged search window or other criteria for correct tracking.

3.6.1 Example WAV's Included with ETD

Like in Chapter 2, a collection of sample WAV files for the results of the correlation technique presented in this chapter has been provided. The samples are for the linear array setup as in Chapter 2 with the same speaker and noise environment and the update threshold chosen for .1, .5, and .9. These files should help demonstrate that the looser thresholds show erratic and quickly degrading performance while the higher threshold, although initially ensuring a good beamform, eventually begins to break down as well. In all cases, it should be clear that, compared to the beamformer output of the traditional GSC as in the included files for Chapter 2, the correlation technique is never as effective.

3.7 Conclusion

In this chapter a method for automatically adjusting the focal point of a beamformer by updating a seed value using a cross correlation technique was presented along with a least-squares method of estimating the focal point of a three-dimensional array given its alignment lags. Results indicate a worsening of performance for all examined scenarios with a steady decline in all cases as the correlation coefficient threshold is

reduced. These results may be due to a bias caused by target and competing speakers never moving and too large of a correlation search window, but may also point toward the general idea that statistical methods may always face serious difficulties under poor SNR conditions.

Chapter 4

Microphone Geometry

4.1 Introduction

In Chapter 1 is was shown that the GSC results from the factoring of the Frost algorithm for an optimal beamformer into two portions: a fixed Delay-Sum Beamformer and an adaptive stage called the Blocking Matrix (BM). Given the fact that results from Chapters 2 and 3 show a clear limit to how much improvement can be realized by improving the adaptive stage, we now turn our attention to the Delay-Sum Beamformer whose performance can be modified only by changing the array geometry. Since equispaced linear arrays are limited in their voice capture capabilities in this chapter we evaluate more general array geometries in two and three dimensions. We begin by introducing visualization of the beamfields with volumetric plots, then go on to analyze stochastic arrays in the general sense through Monte Carlo simulations using a set of proposed evaluation parameters and compare the performance of the irregular arrays to that of a regular rectangular array. Finally we conclude with some guidelines for optimal microphone placement.

4.2 Limitations of an Equispaced Linear Array

The traditional equispaced linear array suffers from three significant problems. The first is that its regular spacing makes it useful only for a narrow range of frequencies. The strongest condition on this range is spatial aliasing, the analog of the Nyquist rate for beamforming which states [4]

$$d < \frac{\lambda_{min}}{2} \tag{4.1}$$

for intermic spacing d. However, the optimal intermic spacing range for a linear array tends to be tighter because as waves are shifted and added together in the DSB both extremes of a relatively long wavelength (not enough change in the shift operation) and relatively short wavelength (too much change in the shift operation) are undesirable. Unfortunately, human speech is an inherently very wideband signal with significant components ranging from 50-8000 Hz [10], indicating that an array tuned to a single frequency will have a far smaller effective bandwidth than necessary.

The second limitation is the fact that an equispaced linear array is steered using only a single parameter θ, the angle of incidence of the target wavefront with respect to the array's axis. This type of steering means that sound sources that are colinear with respect to the array steering cannot be resolved. In addition, the rotational symmetry of the array means that sounds at different heights for a horizontal array cannot be resolved, either.

The third limitation is the fact that under many circumstances an equispaced linear array may not be feasible to construct. For example, in the case of a smart room such as that constructed in the AVA Lab (Ambient Virtual Assistant) at the University of Kentucky Visualization Center, microphones placed in a ceiling are subject to placement constraints such as lighting, ventilation systems, or the metal ceiling tile grid. In the case of a surveillance system an array may need to be placed too quickly and discreetly for precise intermic spacings to be enforced. And even in the event that an equispaced linear array can be constructed precise microphone placement can be very difficult to achieve even with laser systems [20].

Thus in light of these issues, we now wish to analyze arrays of more general geometries to see what layouts might work better for human voice capture. We begin by studying the plot of the sound power that a beamformer picks up as a function of position in space, called the beampattern.

4.3 Generating and Visualizing 3D Beampatterns

The response of a linear array as a function of steering angle is a one-dimensional function of θ, but if we generalize the array and its steering capability to $\mathbb{R}^3$ then we face the challenge of generating a volumetric plot–a visualization of a function of three variables. Here we wish to plot the beamformer power as a function of a Cartesian (x, y, z) coordinate.

Since human perception can understand only three spatial coordinates, we choose to use color as our fourth dimension in the plots. Here we choose to use the classic Jet colormap which renders the weakest intensities in blue and then progresses to green, yellow, orange, and finally red for the strongest intensities. In addition, we recognize that our rendering will require the ability to see into a volume, since areas of low intensity will wrap around areas of high intensity and may obscure our view if great care is not taken. For that our solution is to use an intensity-dependent transparency that renders the weak areas lightly (very transparent) and the strong areas heavily (nearly opaque).

The plots are generated by propagating a burst of noise colored to match the SII spectrum onto an array of microphones using a sound simulator software and evaluating the beamformer response throughout some volume of interest. Since the beamfield must naturally be evaluated at only a discreet number of points, we choose the beamfield resolution as

$$\Delta_{grid} = \frac{.4422c}{f_{max}\sqrt{d}} \tag{4.2}$$

where d is the dimension of the grid space (3 for a volumetric plot) and f_{max} is the

greatest frequency of the target sound. This choice of spacing ensures that no more than a 3 dB change in the beamfield will occur between grid points [19].

The operations of holding a sound source stationary and sweeping the array focal point and holding the focal point stationary and sweeping the sound source position are equivalent operations for generating the DSB beamfield in a small room where, as shown in Chapter 2, sound attenuation through air has a negligible effect over only a couple meters (.6 dB at the highest frequencies, which is significantly smaller than the 3dB threshold of variation for the grid spacing). To see this, consider the fact that for a source at point $\mathbf{a} = (a_x, a_y, a_z)$ the simulated signal $x[n]$ at the i^{th} microphone with position (x_i, y_i, z_i) will be

$$x_i[n] = x[n - \tau_a] \tag{4.3}$$

where

$$\tau_a = \frac{f_s}{c}\sqrt{(x_i - a_x)^2 + (y_i - a_y)^2 + (z_i - a_z)^2} \tag{4.4}$$

and that the delay applied in the DSB operation to find the power at point $\mathbf{b} = (b_x, b_y, b_z)$ is

$$\tau_2 = \frac{f_s}{c}\sqrt{(x_i - b_x)^2 + (y_i - b_y)^2 + (z_i - b_z)^2} \tag{4.5}$$

Thus the DSB computes

$$y[n] = \frac{1}{M}\sum_{i=1}^{M} x_i[n - \tau_b] \tag{4.6}$$

$$= \frac{1}{M}\sum_{i=1}^{M} x[n - \tau_a - \tau_b] \tag{4.7}$$

where clearly the order of delays is irrelevant. This choice in the order of operations is significant because it allows us to run the sound simulator only once rather than at every grid point in the volume of interest, which is a very time consuming operation. (For the current Matlab implementation, this reversal can make the difference of thirty minutes of processing spread out over computer cluster versus ten minutes on a single PC.)

4.4 A Collection of Geometries

In Section 4.3 we outlined our algorithm for visualizing beampatterns in three dimensions. We now display the beampatterns for several of the office setting microphone arrays from the Vis Center data archive for specified focal points in order to gain some insight into what makes for an effective array and what doesn't. Note that all the arrays except for Random Array 1 have the same intensity colorbar scale ranging from -2 to -12 dB below the focal point maximum and that the microphone positions are overlaid as gray dots. Also note that there is no single beampattern for an array (the farfield linear array is the single exception), but as will be shown in the Monte

Carlo experiment the beamfield generated for a source point below the center of the array will be the best case scenario and that array performance should always degrade for all other focal points.

4.4.1 One Dimensional Arrays

4.4.1.1 Linear Array

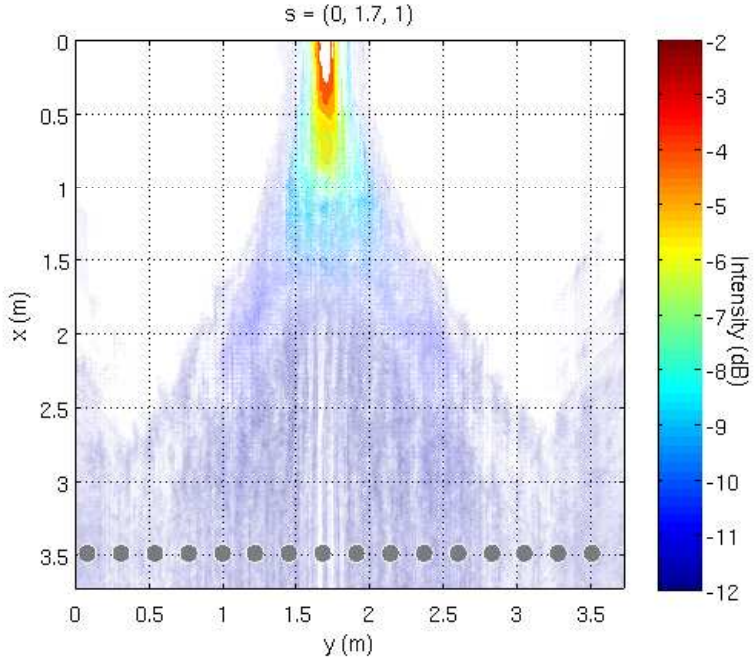

Figure 4.1: Linear Array Beamfield, Bird's Eye View

The linear array, for as much as it's been criticized so far, performs rather well comparatively. Because of the nearfield nature of the array and the fact that the beamfield isn't viewed as a function of angle the traditional sinc pattern isn't readily apparent. One may argue, however, that this fact is an advantage of a linear array in an office environment where the large aperture size of the array relative to the enclosure means that sidelobes will rarely fit inside the room. Notice also that the mainlobe is clearly elongated in the direction of the array and the rotational symmetry of the beampattern in the perspective view. The perspective view of this beampattern is one of several that demonstrates that assuming zero variance in the beamfield with respect to z is a reasonable approximation.

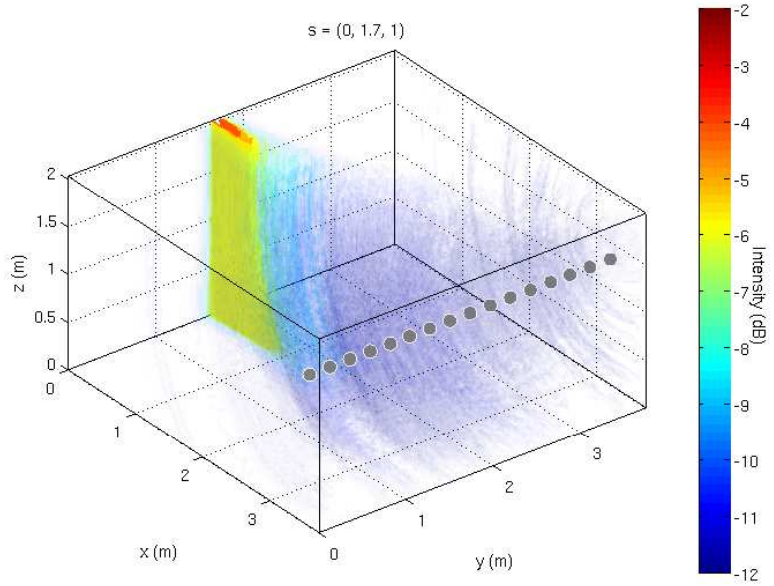

Figure 4.2: Linear Array Beamfield, Perspective View

4.4.2 Two Dimensional Arrays

4.4.2.1 Rectangular Array

The rectangular array has a tighter mainlobe than the linear array, but the bird's eye view shows that the sidelobes are more prominent and radiate out from the mainlobe much further than for the linear array. While the beampattern varies somewhat with height the features show only slow variation in the z direction.

4.4.2.2 Perimeter Array

The perimeter array does a very good job of keeping a tight mainlobe along with nearly uniform suppression everywhere else in the room. There's also very little variance of intensity with height.

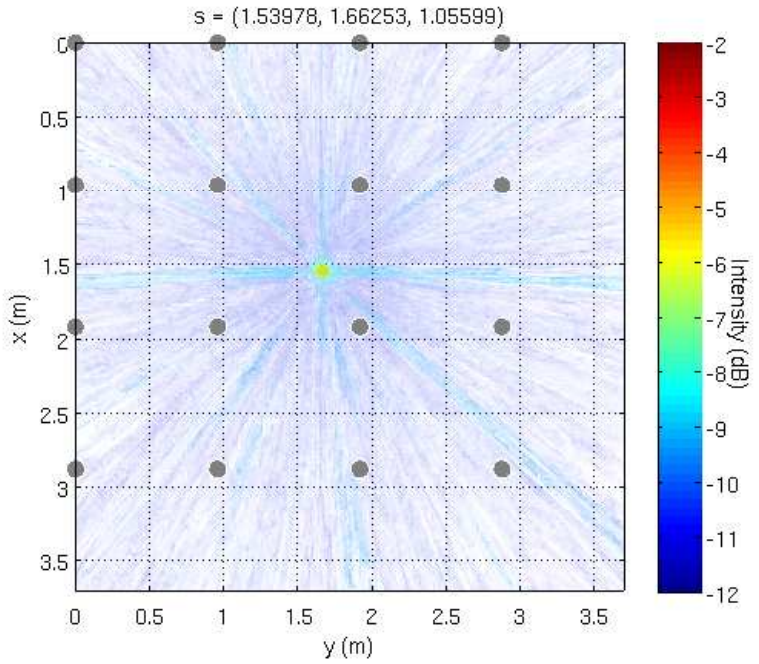

Figure 4.3: Rectangular Array Beamfield, Bird's Eye View

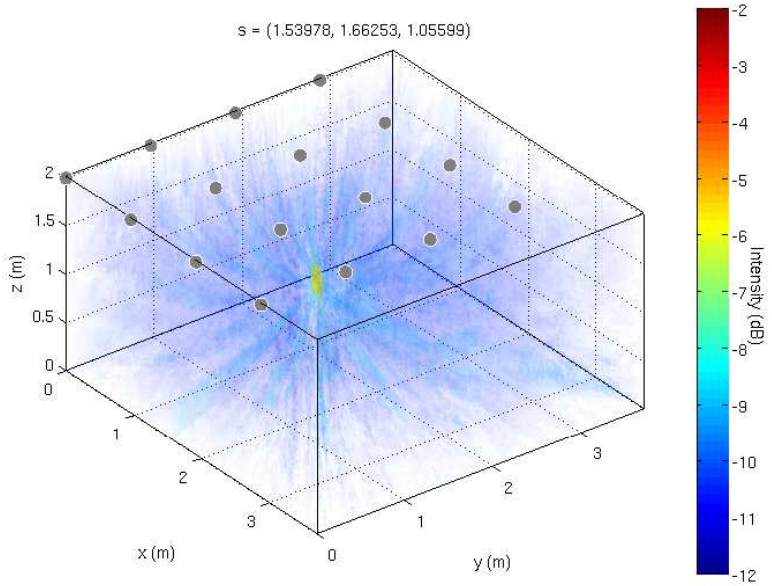

Figure 4.4: Rectangular Array Beamfield, Perspective View

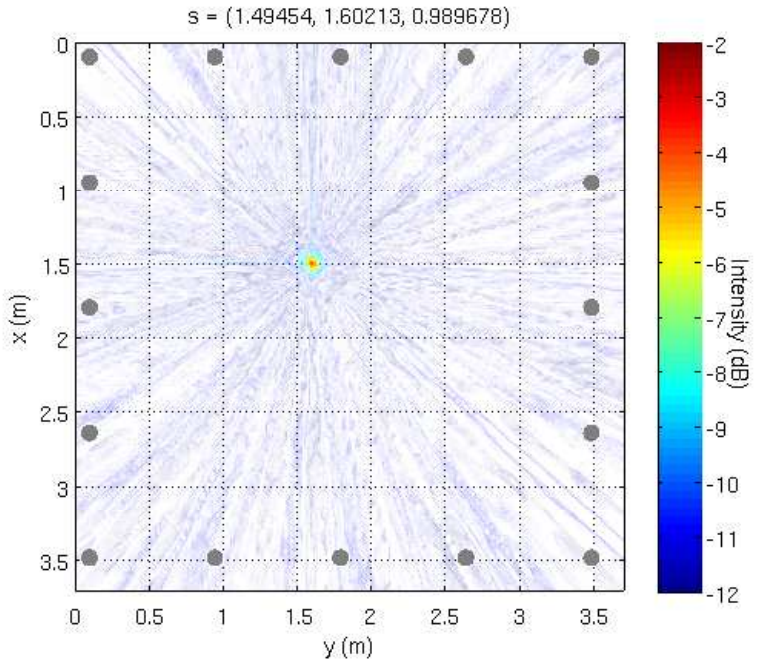

Figure 4.5: Perimeter Array Beamfield, Bird's Eye View

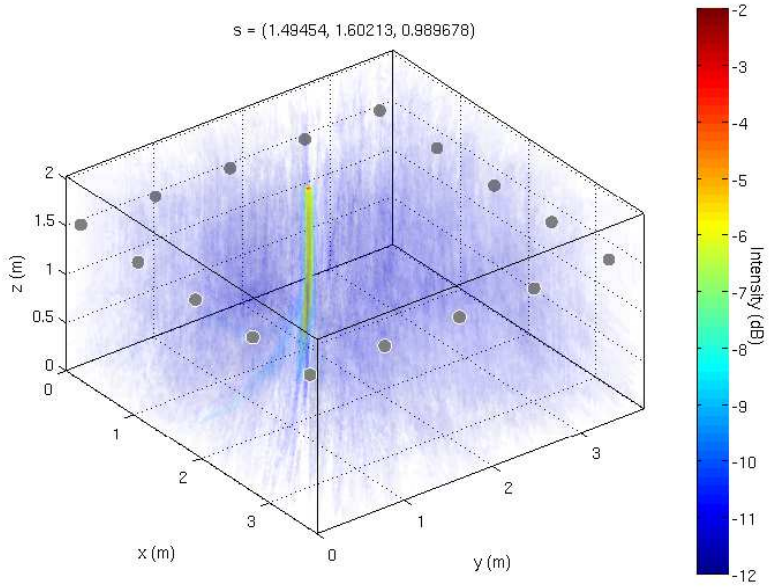

Figure 4.6: Perimeter Array Beamfield, Perspective View

4.4.2.3 Random Ceiling Array 1

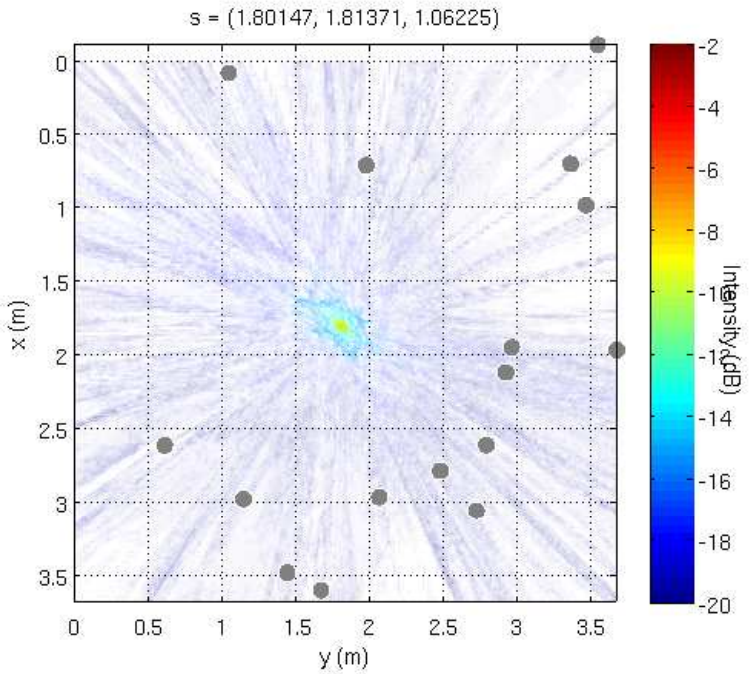

Figure 4.7: First Random Array Beamfield, Bird's Eye View

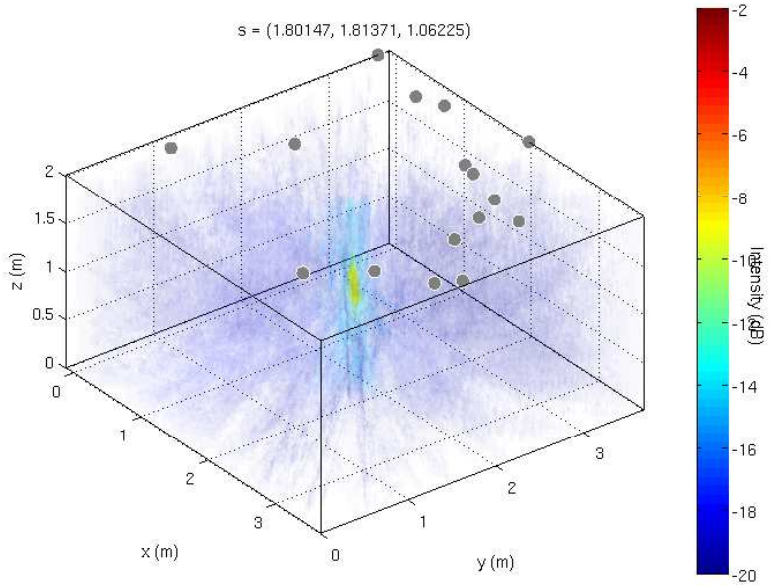

Figure 4.8: First Random Array Beamfield, Perspective View

This first random array (the one used in the experiments in Chapters 2 and 3) has the strongest DSB beampattern of all the arrays to be considered in this section. Outside its mainlobe the suppression is so strong that the color scale has to range down to -20 dB to pick it up (as opposed to -12 dB for all the others). Again, note the small variation in the z direction.

4.4.2.4 Random Ceiling Array 2

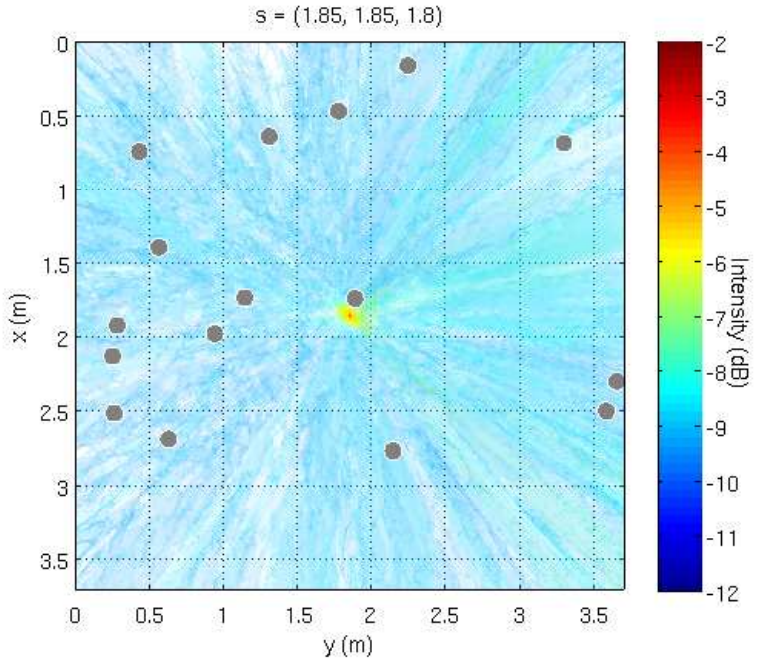

Figure 4.9: Second Random Array Beamfield, Bird's Eye View

This second random array does well to demonstrate that "random" doesn't necessarily mean "effective". This array performs by far the worst out of all those considered, where on average the signal suppression outside the mainlobe is hardly better than -8 dB when nearly all others get down to at least -12 dB. The two random arrays presented here demonstrate that while an irregularly-spaced microphone array shows great potential more work must be done in order to quantify what it is about the "randomness" that translates into better performance.

4.4.3 Three Dimensional Arrays

4.4.3.1 Corner Cluster

The corner cluster array illustrates the extreme form of what happens when an array has a small aperture size and is heavily lopsided away from its target (subjects to be addressed more formally in the next section). The main lobe of the array is both very wide and elongated.

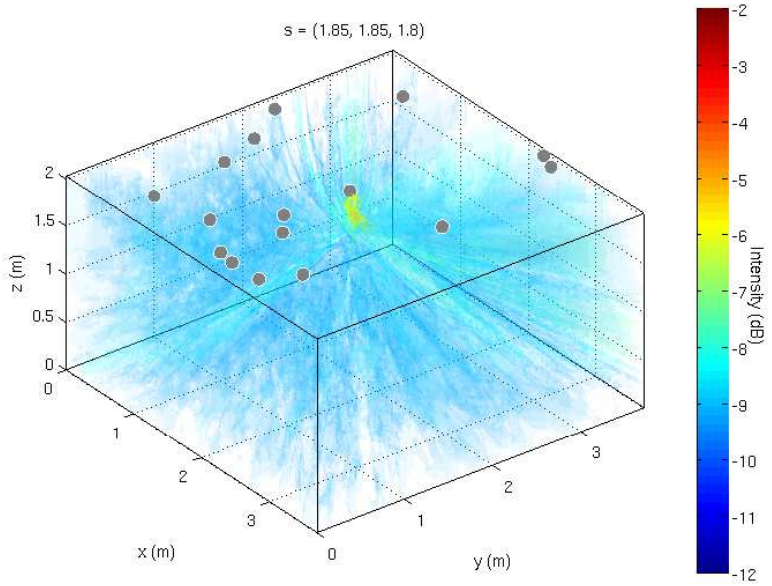

Figure 4.10: Second Random Array Beamfield, Perspective View

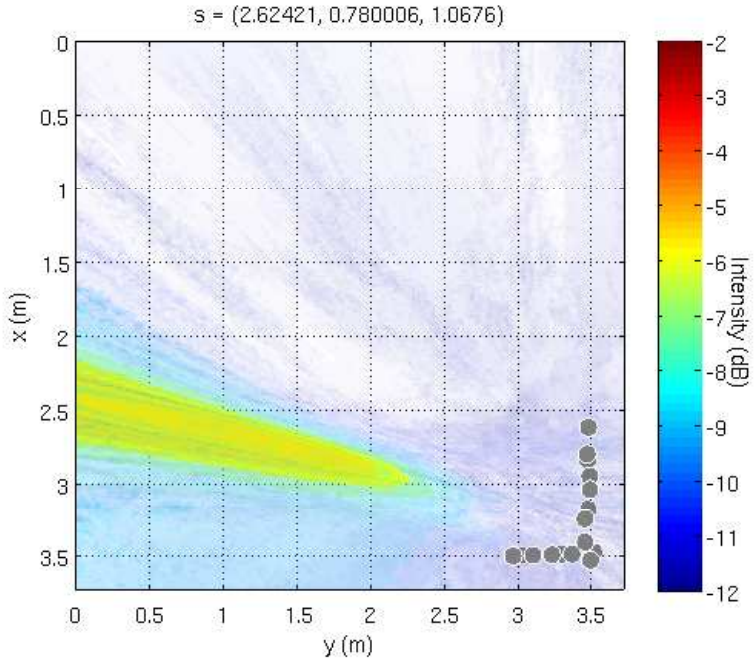

Figure 4.11: Corner Array Beamfield, Bird's Eye View

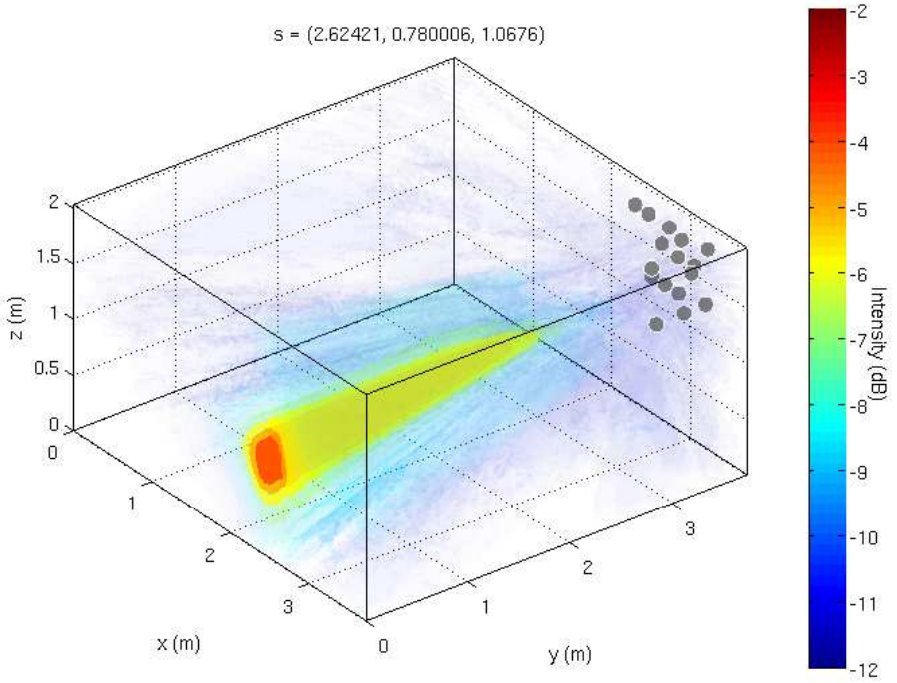

Figure 4.12: Corner Array Beamfield, Perspective View

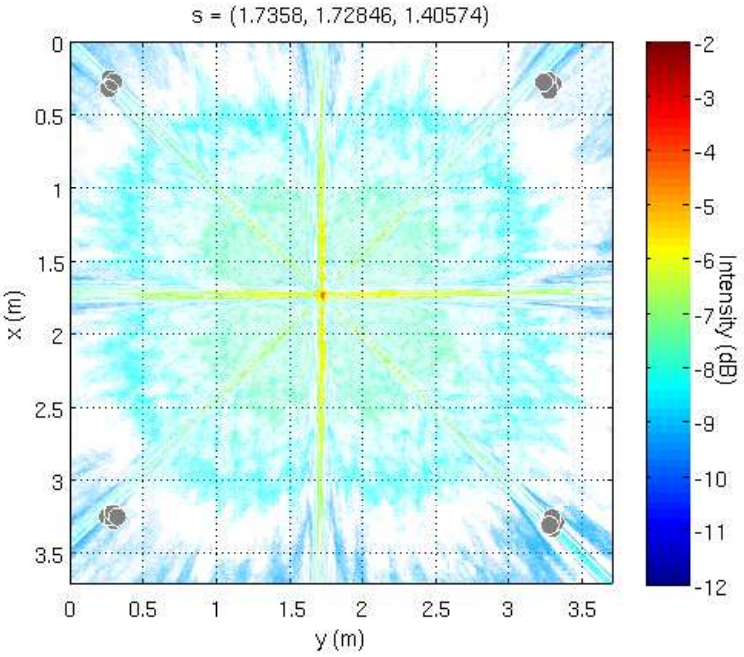

Figure 4.13: Endfire Cluster Beamfield, Bird's Eye View

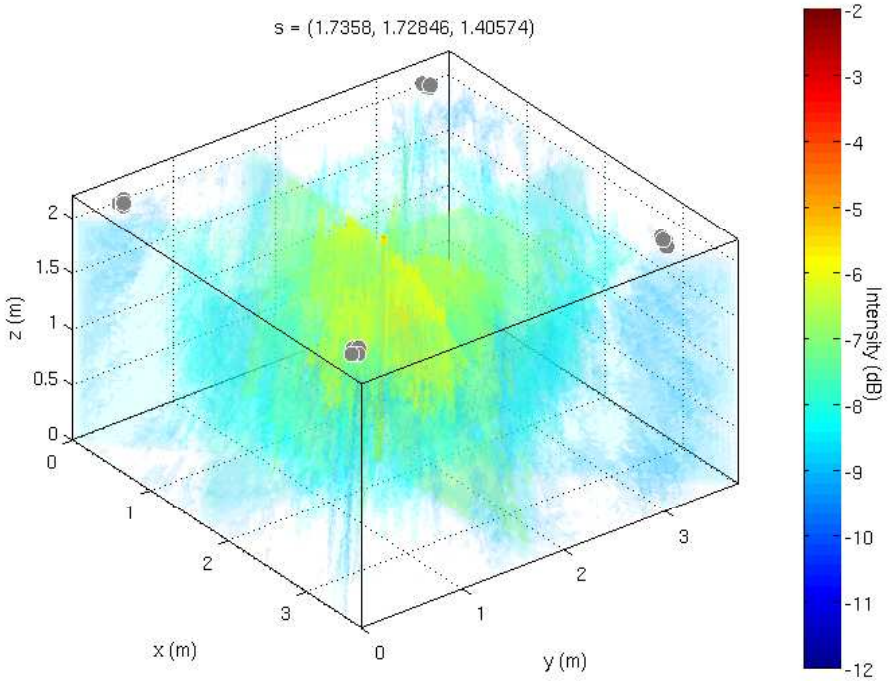

Figure 4.14: Endfire Cluster Beamfield, Perspective View

4.4.3.2 Endfire Cluster

The idea behind the endfire cluster array was to attempt to design an array with clusters of microphones with small intermic spacings that would be optimal for beamforming at high frequencies and then spread the clusters out so that between clusters the beamformer would also be optimized for low frequencies. This hypothesis turns out to be incorrect as one examines the beamfield, where although the mainlobe is very tight sidelobes are very strong and suppression is generally very bad throughout the room.

It's also worth pointing out that although the endfire cluster array is technically 3D the variation in z of its mic positions is small, hence the small variance of its beamfield in the z direction.

4.4.3.3 Pairwise Even 3D Array

The pairwise array is another example of how combining strictly closely and loosely spaced microphones is ineffective at achieving good interference suppression for the DSB. Virtually an entire quarter of the room is part of the mainlobe in these plots.

4.4.3.4 Spread Cluster Array

This array again shows that an irregular array has just as much of a chance of performing poorly as performing well.

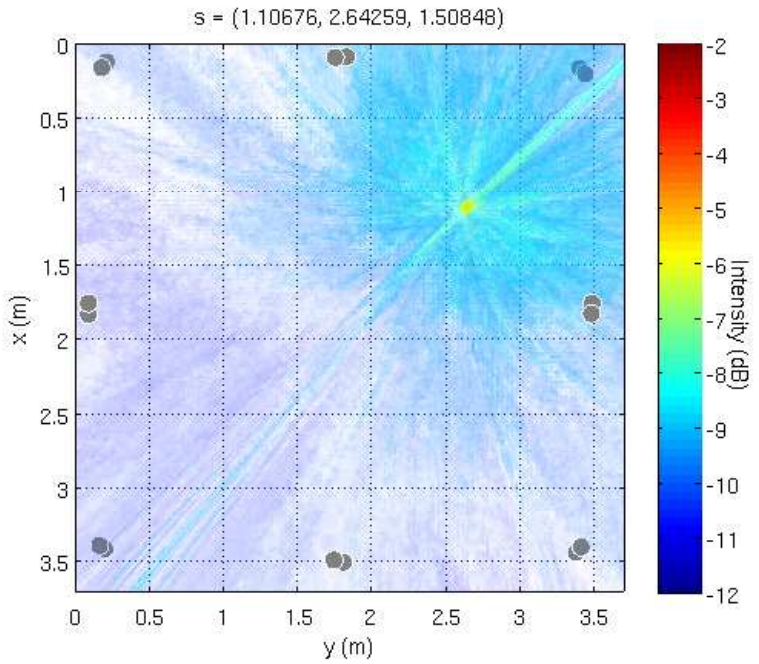

Figure 4.15: Pairwise Even 3D Beamfield, Bird's Eye View

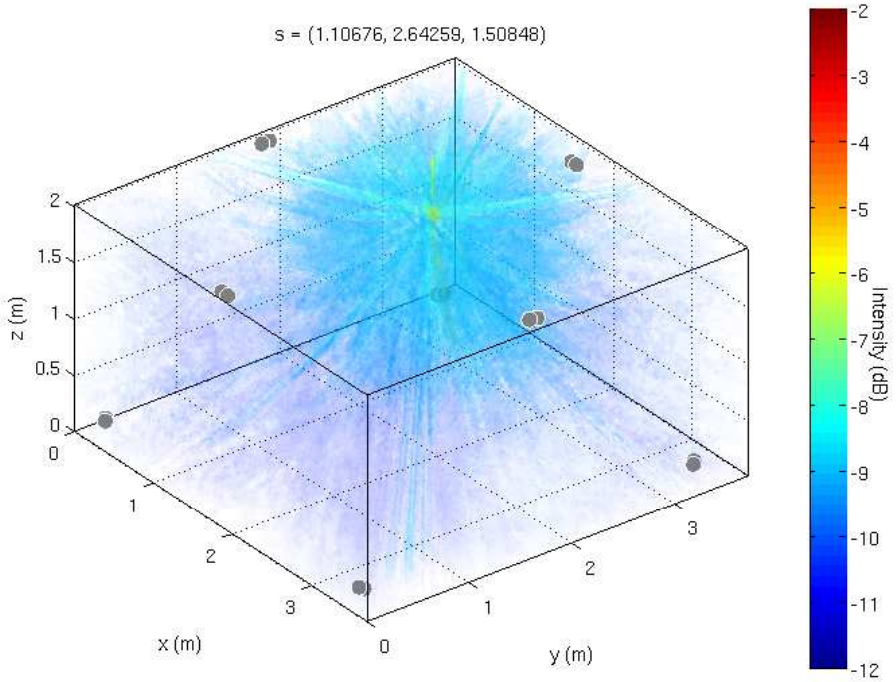

Figure 4.16: Pairwise Even 3D Beamfield, Perspective View

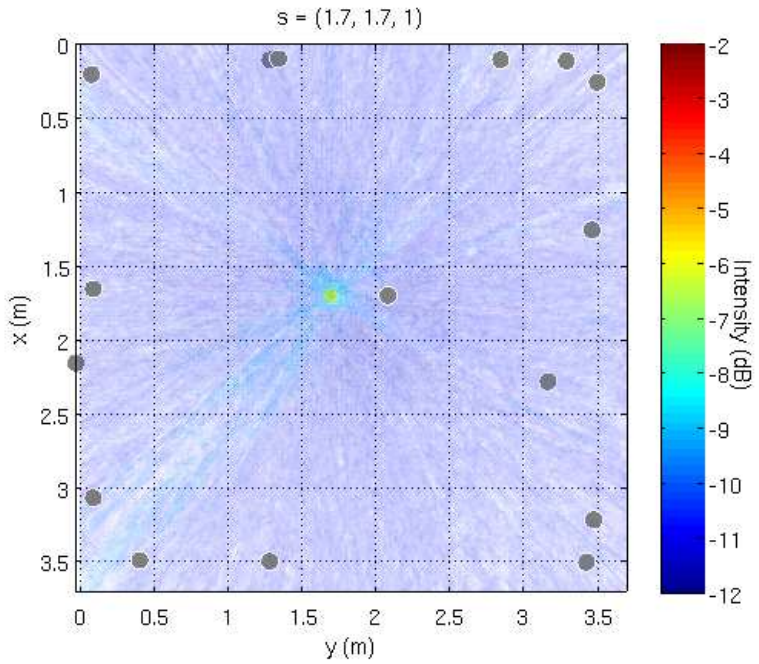

Figure 4.17: Spread Cluster Beamfield, Bird's Eye View

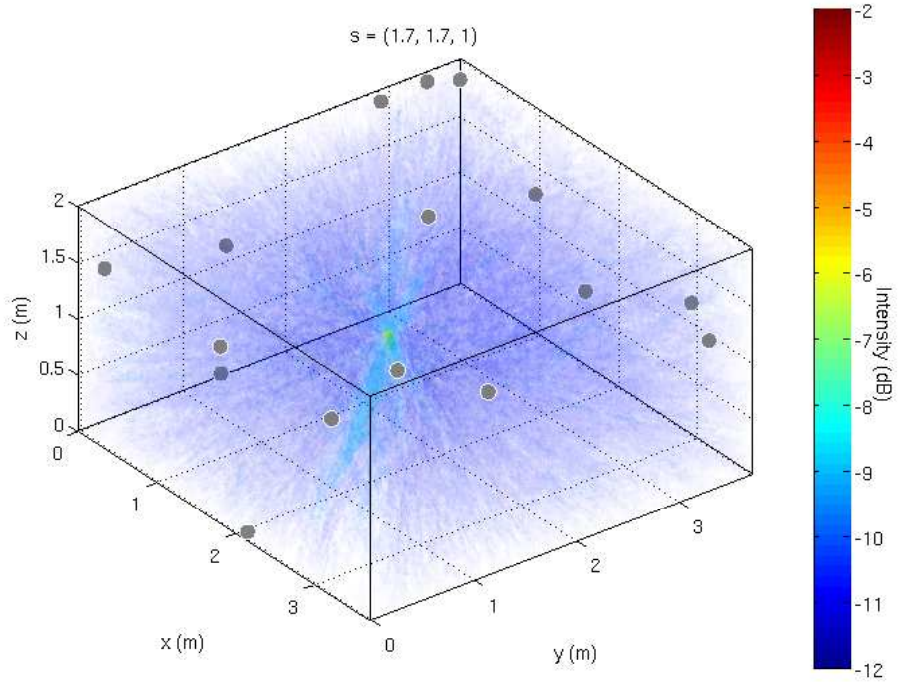

Figure 4.18: Spread Cluster Beamfield, Perspective View

4.4.4 Comparison of Beamfields to Earlier Experimental Results

Now that we have some ability to visualize the DSB beampatterns of the arrays in the data archive, we now compare these plots to the results of Chapters 2 and 3, where almost regardless of algorithm the order of array performance from best to worst was:

1. Linear
2. Random Array 1
3. Rectangular
4. Perimeter

The beamfields presented somewhat agree with this list: the random array should be toward the top of the list and the rectangular toward the bottom. But from beamfields alone we'd expect the perimeter array to beat out the rectangular and the random array to be the best of all, especiallly over the linear array. Three potential explanations for this outcome are:

1. The locations of the target speakers in all tests was well known via Sound Source Location (SSL) but no localization was carried out for the interfering cocktail party recordings. This means that some tests may have been biased toward certain arrays and against others where the sidelobes of an array never had interferers in one experiment but did have them in another. Although the data set used was moderately large the confounding effect may have been signficant. This would be rectified in the data set by forcing all target positions to be measured.

2. The relative performance of the Delay-Sum beamformer in the GSC to the blocking matrix was not assessed. Notice that in the results of Chapter 2 that the Linear array had the best GSC output but also the most leakage in the BM. Since the DSB and BM must work together in the GSC evaluating their relative performance would likely reveal a great deal about array performance, but this analysis is left as future work.

3. The algorithm used for generating the beampatterns made the assumptions that the attenuation of sound through air was negligible and that there was no reverberation at the room boudnaries, both of which may practically be incorrect. More thorough simulation may rectify this potential problem, but the processing time required to generate plots would be very long.

4.5 A Monte Carlo Experiment for Analysis of Geometry

4.5.1 Proposed Parameters

In addition to our more qualitative analysis of visually inspecting and judging beampatterns we'd like to have a quantitative way to compare them, especially if we'd like to compare lots of arrays statistically. Since we've seen that a two-dimensional array can behave as well as a fully 3D one and can be constructed more easily, following parameters for characterization of a 2D beamfield are proposed:

- **Main Lobe Width (MLW)** - The 3 dB width of the main lobe, calculated from the x and y 3 dB widths using

$$w_{3dB} = \sqrt{x_{3dB}^2 + y_{3dB}^2} \tag{4.8}$$

- **Peak to Sidelobe Ratio (PSR)** - The difference between the mainlobe and greatest sidelobe powers in decibels

$$PSR_{dB} = P_{main} - P_{sl} \tag{4.9}$$

The MLW is a measure of the beamform resolution, and the PSR is a measure of the worst-case leakage of noise from other sources outside the main lobe.

4.5.2 Experimental Setup

In order to evaluate the performance of many possible planar array configurations Monte Carlo simulations were performed for a planar array of microphones placed in the ceiling of an 8×8 meter room. In each case the DSB beamfield was computed over a field of view (FOV) in a plane located 1 meter from the origin in the z direction. (Remember, since we've shown that planar arrays tend to show little beampattern variation with respect to height evaluating just a single slice of z will save a lot of processing time.) The focal point was selected as the center of the FOV. With the microphone positions taken as random variables on a uniform distribution for x and y positions, the beampattern and performance metrics were computed. Monte Carlo experiments were run to show relationships between the performance metrics and following two properties:

- Skewedness of the array, defined here as the displacement of the centroid of the array in the xy-plane from the origin. The centroid of the array in x and y is the arithmetic mean of the microphone x and y coordinates, respectively, calculated as

$$C_{x,y} = \left(\frac{1}{M} \sum_{M} x_m, \frac{1}{M} \sum_{M} y_m \right) \tag{4.10}$$

and the displacement of the centroid from the origin is its distance from $(0, 0)$

$$C_r = \sqrt{C_x^2 + C_y^2} \tag{4.11}$$

In order to vary the centroid, four sets of simulations were run where the bounds of microphone selection are constrained to a 2×2 meter box centered at $(0, 0)$, $(1, 0)$, $(2, 0)$, and $(3, 0)$. Over these experiments the frequency and microphone number were fixed at $f = 440$ Hz and $M = 16$. The number of microphones was chosen to reflect the number normally used in the audio data archive while the frequency was chosen low so that the grid spacing could be looser, meaning less processing time and more data points gathered.

- Microphone spread, defined here as the radial dispersion of the microphones. Dispersion in x and y is calculated as the standard deviation of the microphone x and y coordinates about the centroid

$$D_{x,y} = \left(\sqrt{\frac{1}{M} \sum_M (x_m - C_x)^2}, \sqrt{\frac{1}{M} \sum_M (y_m - C_y)^2} \right) \tag{4.12}$$

and define a radial dispersion as

$$D_r = \sqrt{D_x^2 + D_y^2} \tag{4.13}$$

In order to vary the dispersion in the Monte Carlo experiment, four sets of simulations were run where the bounds of microphone selection are centered at the origin but expanded in both x and y from $[-1/4 \;\; 1/4]$ to $[-1/2 \;\; 1/2]$, $[-1 \; 1]$, $[-2 \; 2]$, $[-3 \; 3]$ and $[-4 \; 4]$. As in the previous case, the frequency and microphone number was fixed ($f = 440$ Hz and $M = 16$).

For each of the situations described above Monte Carlo computer simulations of 120 runs were conducted where microphone positions were randomly selected on the appropriate interval. A sound simulator created the array recordings without room reverberations but this time with frequency-dependent acoustic attenuation. A single frequency target signal was used located at (0, 0, 1). Finally, the DSB algorithm was applied to each grid point in the $z = 1$ plane, yielding the slice of the array beamfield.

4.5.3 Results

Plots of the Monte Carlo simulation results are shown in Figures 4.19 and 4.20. The data are represented as error bar plots due to the statistical nature of the experiment, where in each case the mean is labeled by a square and the error bars span ± one standard deviation about the mean. Several of the MLW experiments resulted in little to no error bar span because the relationship in Eq (4.2) allows for relatively loose bounds on grid spacing while ensuring a not-too-large change in beamfield intensity (as much as 10 cm for 440 Hz). In addition, performance metrics for a regularly-spaced

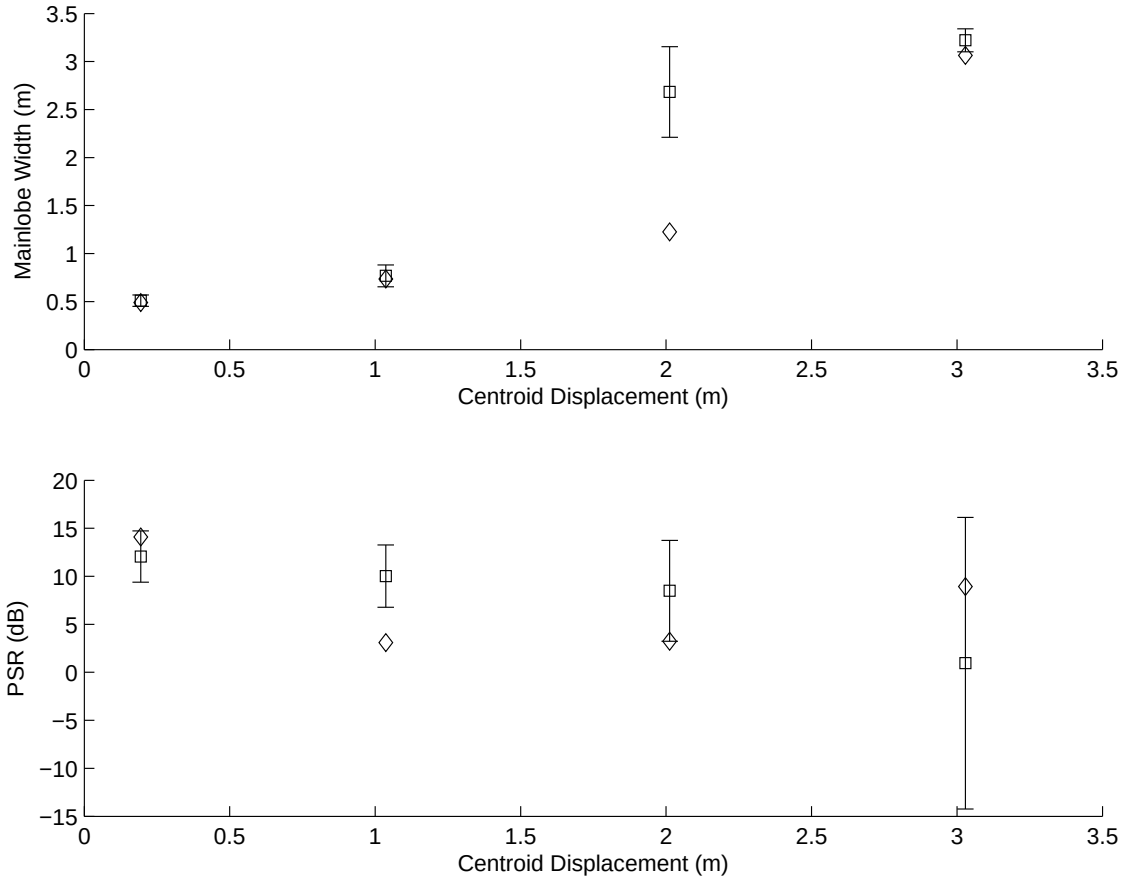

Figure 4.19: Error Bar Plot for Varying Array Centroid Displacement.

rectangular array under similar conditions (same number of microphones, centroid position, dispersion, and/or target frequency as appropriate) were also computed and overlaid as diamonds.

Now examining Figure 4.19 one notices that increasing centroid displacement is harmful in both ways: as the array becomes more lopsided both the MLW grows greatly and the PSR drops substantially. These degradations are not explained by a change in aperture size and intermic spacing but are explained by examining the corner array analyzed earlier. As all members of the array are displaced further from the beamform target less directional resolution is possible, causing the MLW to grow. This lack of resolution also allows partial coherences to become more pronounced, causing the PSR to fall as well. One also finds that the regular array perform worse here, providng further evidence that irregular geometries provide robustness to strong partial coherences.

On the other hand, as the dispersion rises in Figure 4.20 the aperture size grows, causing MLW to drop, but the intermic spacing also grows, causing PSR to drop. The regular arrays generally out-perform the stochastic ones, though it's worth pointing out that the regular array is again centered over its target, which we've shown is its optimal placement.

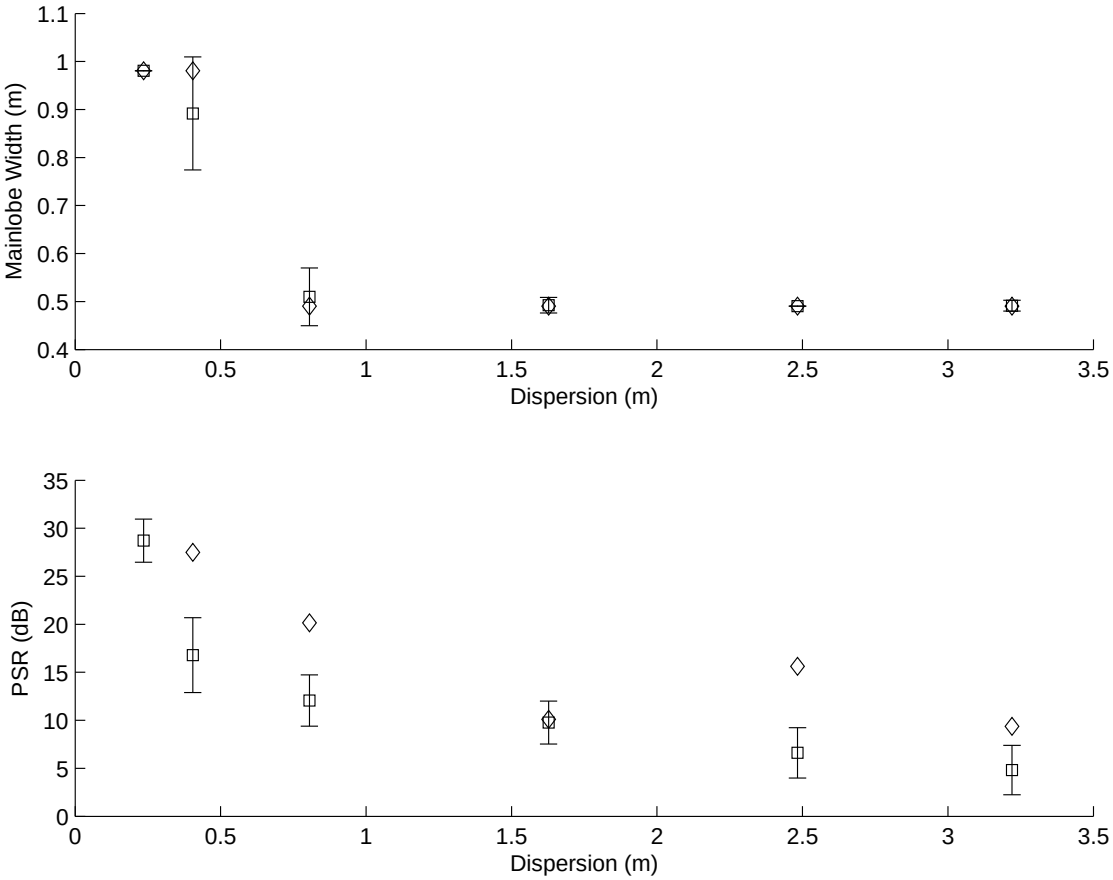

Figure 4.20: Error Bar Plot for Varying Array Dispersion.

What this experiment has shown is that the more centered an array is over its target the better it will perform, both in mainlobe width and peak to sidelobe ratio. The dispersion experiments, however, show more of a tradeoff relationship between MLW and PSR as the overall spread of the array increases. This experiment is a first step in the direction of finding the best statistical parameters for predicting array performance.

4.6 Guidelines for Optimal Microphone Placement

For a fixed number of microphones in an array, the following recommendations for microphone placement can be made given the investigations carried out in this chapter:

- Regular arrays tend to have stronger mainlobes than irregular ones but have stronger sidelobes, too. Hence an irregular array is more robust to changing target and interferer positions.

- A ceiling array tends to work just as well as a fully three-dimensional array assuming that no height resolution is required. In a typical office setting this is a reasonable assumption.
- The more centered a ceiling array is over its target the better, so if the array is used to listen at potentially any point in the room its centroid should be at the center so that no one listening spot is unusually bad.
- Since regularly-spaced arrays are optimal only for a limited range of of frequencies but human voice is an inherently broadband signal an irregular array with a variety of intermic spacings will tend to work better. It's important to note, however, that what makes for an optimal selection of intermic spacings has yet to be determined.

It's worth pointing out that we state "for a fixed number of microphones" because one of the few surefire ways to improve array performance is with more microphones, where it's been observed that every doubling of the number of mics yields about a 3 dB improvement in DSB performance [21].

4.7 Conclusions

In this chapter we described how one can visualize the beampattern for an array as a volumetric plot in three dimensions and used this ability to examine the DSB performance of many of the arrays in the Audio Data Archive and compare this to the experimental results from Chapters 2 and 3. Our qualitative analysis of these plots showed that regularly-spaced arrays can perform well but can be subject to strong sidelobes and that irregular arrays have the potential to overcome this weakness. We also found that one and two-dimensions arrays tend to show little variation in their response with respect to height and that the 3D arrays examined offered little performance benefits over the planar arrays, giving us motivation for conducting further analysis for only the somewhat simpler 2D case.

In an attempt to create qualitative means with which to compare arrays we defined mainlobe width and peak to sidelobe ratio in two dimensions and tested how these parameters changed for varying centroid and dispersion of a planar array in order to evaluate arrays in a statistical sense. Here we found that a ceiling array performs best when centered over its target and that a tradeoff relationship exists between PSR and MLW as the dispersion, or "spread", of an array is increased.

The open question remains–what is it that makes one irregular array "good" and another "bad"? The answer is likely the most important future work that could come from the current project, and one that would likely involve computing histograms of intermic/interpath spacings and finding the optimal statistics.

Chapter 5

Final Conclusions and Future Work

The goal of this thesis was to improve upon the Generalized Sidelobe Canceller algorithm for its use in an immersive office setting. The analyses of Chapters 2, 3, and 4 have yielded the following broad conclusions:

- Purely statistical methods for improving performance, such as the expected value amplitude scaling in Chapter 2 and correlative steering in Chapter 3, are ineffective when the SNR is low. Since one would beamform only when the SNR is low, those methods must be deemed ineffective.
- The more relevant information that can be collected independently of the raw audio tracks the better. A tiny bit of performance gain was realized in Chapter 2 by using an acoustic physics-based filter that required measuring the temperature, humidity, and atmospheric pressure of the office, while in Chapter 3 the best performance was realized when the beamformer remained steered at its original seed focal point.
- Experimentally the linear array did best, but an irregular planar ceiling array did almost as well, was less susceptible to bad speaker arrangement, and was easier to construct.
- A ceiling array performed best when its centroid (mean microphone position) was centered over its target.

The amplitude correction attempted in Chapter 2 was never really able to do much better than the traditional Griffiths-Jim method of simply taking differences of tracks. A method using an ISO-based filter offered a very small benefit, while a statistical scaling algorithm did the worst out of the lot. The most significant point from this chapter, however, was that even when a theoretically perfect BM was used the array geometry was the dominant factor in determining array performance, suggesting that future research into optimal microphone arrangement would be the most fruitful.

The cross correlation technique in Chapter 3 also followed the trends of Chapter 2, where as the correlation threshold was loosened and the beamformer gained more decision-making ability performance degraded significantly. But here too, regardless of the threshold used, array geometry was still most important in determining how well the beamformer performed. The multilateration technique presented in that chapter was a useful debugging tool for visualizing how the beamformer behaved, especially since it alone showed that the windowing for the correlation search window would need to be tightened in order to realize an overall limit for potential focal point shift since, in the worst case, many of the lags can change by their prescribed maximua and compound into an overall shift much larger than intended.

The analysis of microphone geometry in Chapter 4 began with a qualitative assessment of the three-dimensional beampatterns of many of the arrays implemented in the University of Kentucky Visualization Center's Audio Data Archive. This analysis showed that, in terms of the Delay-Sum beamfield alone, there exists the potential for an irregularly-spaced array to do far better than any of the traditional regularly-spaced ones, but the open questions remain as to how a DSB beamfield translates into overall GSC performance and what sorts of measures could directly indicate which irregular arrays would be effective and which ones wouldn't. As a start toward applying statistical parameters on array geometry, the performance of planar arrays was evaluated with a Monte Carlo simulation by testing how variations in array centroid and dispersion affected mainlobe width and peak to sidelobe ratio. Here it was found that a centroid closer to focal point always yielded better results while the dispersion showed a tradeoff relationship between MLW and PSR where a small dispersion meant a wide mainlobe but weak sidelobes.

It's worth emphasizing one more time that array geometry is, by far, the dominant factor in determining array performance. After all, remember that the original optimization problem as proposed by Frost involved getting the best performance out of an array where the array layout was viewed as a constant, not a variable. Hence future work into immersive beamforming should focus on optimal microphone positioning and finding better parameters that can determine how this optimization should be carried out.

Appendix A

Stability Bounds for the GSC

A.1 Introduction

The Generalized Sidelobe Canceller (GSC) is an adaptive noise cancellation system used in array processing systems [7] [9]. The noise reference signal is obtained from the array signals by effectively nulling out the desired signal via a blocking matrix and filtering the array channels with adaptive normalized least mean square (NLMS) filters. While parameter values for the adaptive algorithm have been suggested for stable operation [5], no general relationship between the system and adaptation parameters has been presented. Hence this relationship is derived here and the stability limits are verified with a series of applications to multichannel speech data recorded in a cocktail party environment.

The GSC output is computed from the M array channels with the following equation:

$$y[n] = b[n] - \sum_{k=1}^{M-1} \mathbf{w}_k^T[n]\mathbf{z}_k[n] \qquad M > 1 \tag{A.1}$$

where $b[n]$ is the fixed beamformer output, $\mathbf{w}_k[n]$ is the k^{th} vector of adaptive filter weights for each output of the blocking matrix, and $\mathbf{z}_k[n]$ is the k^{th} blocking matrix output tracks window of length O. The filter weights for all output channels are updated using:

$$\mathbf{w}_k[n+1] = \beta\mathbf{w}_k[n] + \mu y[n]\frac{\mathbf{z}_k[n]}{||\mathbf{z}_k[n]||^2} \qquad 1 \leq k < M \tag{A.2}$$

where β is the forgetting factor $(0 < \beta < 1)$, $||\cdot||^2$ is the squared Euclidean norm, and μ is the step size parameter ($\mu > 0$). The parameter μ determines the magnitude of the filter tap changes every iteration. Large values of μ result in rapid convergence toward a steady-state signal with large misadjustment (variations around the ideal Wiener filter taps), whereas small values result in slow convergence with small misadjustment. The forgetting factor β affects the influence of previously calculated tap weights on the future weights. Choosing $\beta < 1$ is useful in nonstationary environments such as audio beamforming where the signal and noise properties vary over

time, and it also provides some robustness for finite precision implementation since it limits the accumulation of quantization errors from previous calculations [22].

The tap update algorithm for the GSC is similar to the classical NLMS algorithm, which has form:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu \frac{\mathbf{u}[n]}{||\mathbf{u}[n]||^2} e[n] \tag{A.3}$$

where the error $e[n]$ given by:

$$e[n] = d[n] - \mathbf{w}^T[n]\mathbf{u}[n] \tag{A.4}$$

with desired response $d[n]$ and vector of input samples $\mathbf{u}[n]$. In the algorithm's original formulation, the tap weights are updated toward a desired response $d[n]$ by following a gradient descent in the direction of $e[n]$. In the GSC, however, the error signal is replaced with the current beamformer output since a desired signal cannot be obtained for the beamformer output. The objective is to minimize the output energy based on the assumption the noise reference signal is not correlated with the target signal, which was removed from the array channels via the blocking matrix [7] [9].

It has been shown [22] that stability for the generic NLMS algorithm is ensured if the step-size is bounded as:

$$0 < \mu < 2. \tag{A.5}$$

In practice, however, it has been encountered that stability in the GSC requires μ to be much smaller [5]. A comparison between the traditional NLMS and GSC update equations reveals that the GSC update has an additional feedback loop over the traditional NLMS: the GSC is recursive in both its tap weights and reference signal, whereas the NLMS is recursive only in its taps. This additional feedback loop suggests an explanation for why μ must be smaller for stability. The explicit relationship between μ, β and M for stability is derived in the next section.

A.2 Derivation

The z-transforms of the output and update equations (Eqs. (A.1) and (A.2)) for the GSC are:

$$Y(z) = B(z) - \sum_{k=1}^{M-1} \mathbf{W}_k^T(z) * \mathbf{Z}_k(z) \tag{A.6}$$

and

$$z\mathbf{W}_k(z) = \beta \mathbf{W}_k(z) + \mu Y(z) * \mathbf{Z}_k(z) * Q_k(z), \tag{A.7}$$

where * represents convolution in the z-domain and

$$Q_k(z) = Z\left\{ \frac{1}{||\mathbf{z}_k[n]||^2} \right\}. \tag{A.8}$$

¿From Eq (A.7) an expression for the tap weights can be written as:

$$\mathbf{W}_k(z) = \frac{\mu}{z-\beta} Y(z) * \mathbf{Z}_k(z) * Q_k(z), \tag{A.9}$$

and substituting this result into Eq (A.6) yields

$$Y(z) = B(z) - \frac{\mu}{z-\beta} Y(z) * \left(\sum_{k=1}^{M-1} \mathbf{Z}_k^T(z) * \mathbf{Z}_k(z) * Q_k(z) \right). \tag{A.10}$$

This equation can be simplified by taking the multiple convolutions inside the summation back in time domain, where the convolutions become multiplications. Given the norm is of the form

$$||\mathbf{x}||^2 = \mathbf{x}^T\mathbf{x}, \tag{A.11}$$

the following simplification results:

$$\mathbf{Z}_k^T(z) * \mathbf{Z}_k(z) * Q_k(z) = Z\left\{ \frac{\mathbf{z}_k^T[n]\mathbf{z}_k[n]}{||\mathbf{z}_k[n]||^2} \right\} = \delta(z), \tag{A.12}$$

where $\delta(z)$ is the Dirac delta function. Substitute this result into Eq (A.10) to obtain:

$$Y(z) = B(z) - \frac{\mu}{z-\beta} Y(z) * \left(\sum_{k=1}^{M-1} \delta(z) \right) \tag{A.13}$$

$$= B(z) - \frac{\mu}{z-\beta} Y(z) * \Big((M-1)\delta(z) \Big) \tag{A.14}$$

$$= B(z) - \frac{\mu(M-1)}{z-\beta} Y(z). \tag{A.15}$$

Stability for the entire adaptive beamformer can be analyzed via the transfer function $H(z)$, where $B(z)$ is the input to the adaptive system and $Y(z)$ the output. This transfer function can be derived from Eq. (A.15) to yield:

$$H(z) = \frac{1-\beta z^{-1}}{1 - \Big(\beta - \mu(M-1)\Big) z^{-1}}. \tag{A.16}$$

Hence stability requires

$$\Big|\beta - \mu(M-1)\Big| < 1 \tag{A.17}$$

This result is consistent with the limits for the classical NLMS given in Eq (A.5), where if $\beta = 1$ (no forgetting factor) and $M = 2$ (only one adaptive filter) then the limits for stability are $|1-\mu| < 1$, which holds only for $0 < \mu < 2$. It is also worthwhile to note that this stability equation makes no assumptions upon the nature of $b[n]$ or any $z[n]$, meaning that changes to the computation of the fixed beamformer or the blocking matrix tracks will not affect the stability of the system with respect to choices of μ and β.

A.3 Computer Verification

In order to examine the result of Eq. (A.17) a 20 second, 16 channel audio recording was made of a target speech signal with interfering speech sources at off-target locations. This created a nonstationary noise and target signal scenario to test for stability of the algorithm over values of β and μ. The number of array channels, M, was effectively varied by taking subsets of the 16 channel recording. Combinations leading to instability were identified when the output grew without bound. Results of a few of these experiments are shown in Figs A.1, A.2, and A.3 for the number of microphone channels M equal to 2, 3 and 4, respectively.

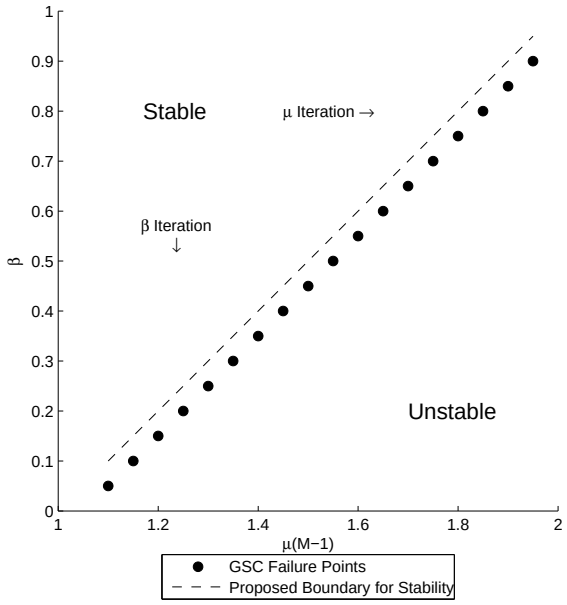

Figure A.1: GSC Stability Plot, $M = 2$, $\beta_{max} = .95$, Voice Input

Each plot was generated by iterating μ in the positive direction for $1 < \mu(M-1) < 2$ with increment $\Delta\mu = .05$. Likewise β was iterated in the negative direction from .95 to 05 in increments of $\Delta\beta = .05$, and the point at which instability occurred for a fixed M was recorded. A run was deemed stable if it ran for 10,000 iterations without failing. The order of the filters was chosen as 20 to save on running time, although filter orders of as much as 256 [23] have been used in practice.

Of particular interest is the boundary in the β-μ plane where instability occurred. The boundary, highlighted by the solid dots, is the first instance of instability for β and μ combinations where the increment direction is as indicated in the figures, starting in the stable region and moving toward unstable. The boundary predicted by Eq. (A.17) is also plotted as a dashed line for comparison. The horizontal axes for all plots were normalized by $M-1$ to illustrate that $M-1$ is the correct scaling for μ and for ease of comparison.

Since each plot records when the threshold of stability has been crossed we expect

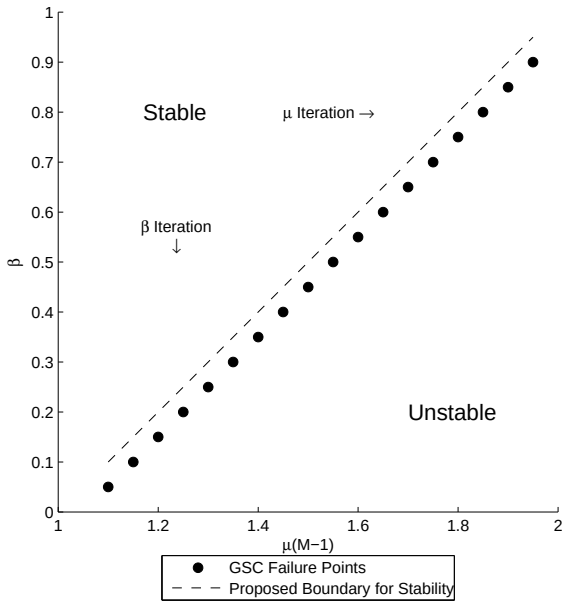

Figure A.2: GSC Stability Plot, $M = 3$, $\beta_{max} = .95$, Voice Input

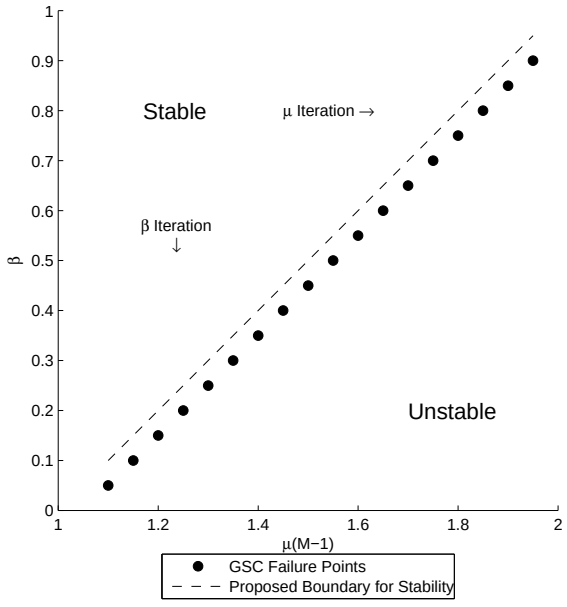

Figure A.3: GSC Stability Plot, $M = 4$, $\beta_{max} = .95$, Voice Input

that with μ and β iterating from top-left to bottom-right the line will be slightly lower than the true boundary due simply to the resolution of iteration. Because of the $M - 1$ scaling, each plot displays a linear pattern rising with a slope of 1. Note the results are consistent with the relationship derived in Eq. (A.17).

A.4 Discussion

The z-transform analysis predicts a linear boundary for stability between μ and β and the experiments on nonstationary noise and target signals agree with this relationship for our chosen values of β and μ. However, in each of these plots β was iterated down from .95, not 1. If we allow $\beta = 1$ then the system can become unstable before reaching the proposed boundary as shown in Figure A.4. Note for $\mu(M-1) > 1.25$ the system became unstable at $\beta = 1$.

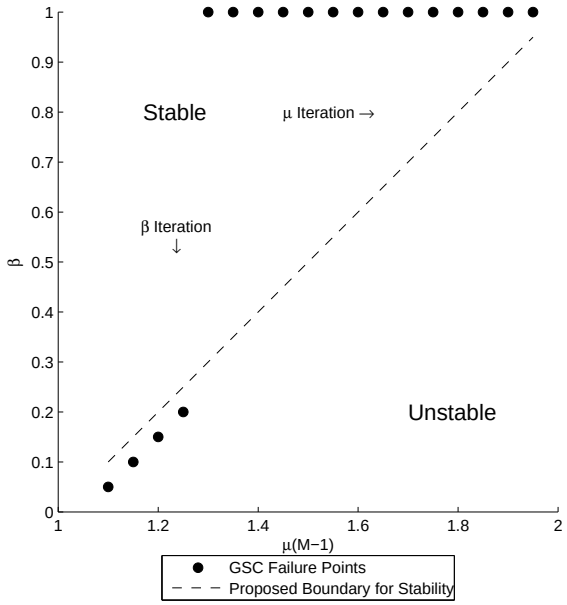

Figure A.4: GSC Stability Plot, $M = 4$, $\beta_{max} = 1$, Voice Input

These failures occur even when a small constant is added to the denominator of the NLMS algorithm to avoid the potential numerical problem of dividing by a small norm if $\mathbf{u}[n] \rightarrow \mathbf{0}$ [22]

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu \frac{\mathbf{u}[n]}{\xi + ||\mathbf{u}[n]||^2} e[n] \qquad \xi > 0 \tag{A.18}$$

One of the intentions of the forgetting factor is to allow the beamformer to track changes in a nonstationary environment without being weighed down by past statistics. If the system is excited with a stationary colored noise the proposed bound will again form, even if we allow for $\beta = 1$ as in Figure A.5. Therefore we conclude that the instability for $\beta = 1$ is due to the algorithm's inability to adapt in a nonstationary environment. Though the forgetting factor for the NLMS has not always been used in the GSC, in its original implementation a small amount of white noise was added to the system [9] which has the same effect as a non-unity forgetting factor [22]. Also, as shown in Figure A.5 for $\beta = 1$, the system can be stable for a limited range of μ.

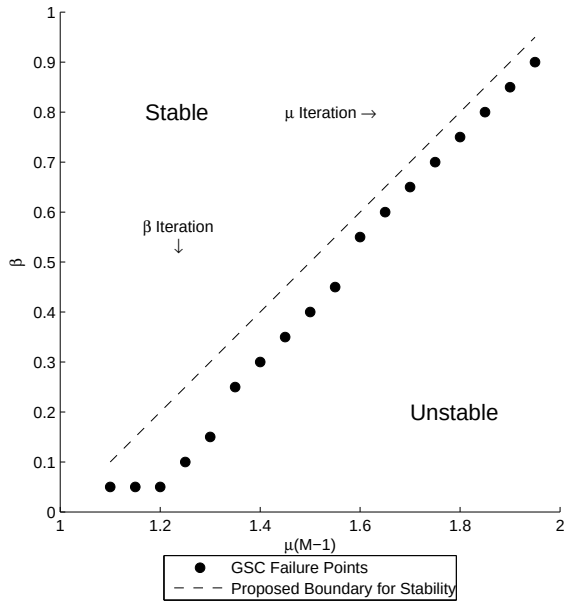

Figure A.5: GSC Stability Plot, $M = 4$, $\beta_{max} = 1$, Colored Noise Input

A.5 Conclusion

It has been shown here via a z-transform analysis that the stability region for the GSC with respect to the choice of μ and β is $|\beta - \mu(M-1)| < 1$. Computer simulations agree with this statement except for the case when $\beta = 1$ where the algorithm fails because it cannot adapt in a nonstationary environment.

Bibliography

[1] Darren B Ward, Rodney A Kennedy, and Robert C Williamson. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 1. Constant Directivity Beamforming. Springer-Verlag, 2001.

[2] Gray W Elko. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 17. Future Directions for Microphone Arrays. Springer-Verlag, 2001.

[3] Microsoft Corporation. Microphone array support in windows vista. *http://www.microsoft.com/whdc/device/audio/micarrays.mspx*.

[4] I.A. McCowan. *Robust Speech Recognition using Microphone Arrays*. PhD thesis, Queensland University of Technology, Australia, 2001.

[5] Osamu Hoshuyama and Akihiko Sugiyama. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 5. Robust Adaptive Beamforming. Springer-Verlag, 2001.

[6] Joerg Bitzer and K Uwe Simmer. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 2. Superdirective Microphone Arrays. Springer-Verlag, 2001.

[7] Otis Lamont Frost III. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60(8):926–935, August 1972.

[8] James Stuart. *Calculus*. Brooks/Cole Publishing Company, 3^{rd} edition, 1995.

[9] Lloyd J Griffiths and Charles W Jim. An alternative approch to linearly constrained adaptive beamforming. *IEEE Transactions on Antennas and Propagation*, AP-30(1):27–34, January 1982.

[10] American National Standards Institute, New York. *ANSI S3.5-1997: American National Standard Methods for Calculation of the Speech Intelligibility Index*, 1997.

[11] Kevin Donohue. Kevin d donohue homepage and experimental data archive. *http://www.engr.uky.edu/ donohue/*.

[12] Wolfgang Herbordt. *Sound Capture for Human Machine Interfaces: Practical Aspects for Microphone Array Signal Processing*. Springer-Verlag, 2005.

[13] Ben Gold and Nelson Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Inc, 2000.

[14] Lawrence E Kinsler, Austein R Fret, Alan B Coppens, and James V Sanders. *Fundamentals of Acoustics*. John Wiley & Sons, Inc, 4th edition, 2000.

[15] K Sam Chanmugan and A M Breiphol. *Random Signals: Detection, Estimation and Data Analysis*. John Wiley and Sons, 1^{st} edition, 1988.

[16] Jospeh H DiBiase, Harvey F Silverman, and Michael S Brandstein. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 8. Robust Localization in Reverberant Rooms. Springer-Verlag, 2001.

[17] Charles Knapp and Clifford Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24(4):320–327, August 1976.

[18] Anand Ramamurthy. Experimental evaluation of modified phase transform for sound source detection. Master's thesis, University of Kentucky, November 2007.

[19] K D Donohue, J Hannemann, and H G Dietz. Performance of phase transform for detecting sound sources with microphone arrays in reverberant and noisy environments. *Elsevier Science Direct*, 2007.

[20] A Muthukumarasamy and K D Donohue. Implact of microphone placement errors on speech inteligibility. *IEEE Southeastcon*, 2008.

[21] M L Seltzer and B Raj. Calibration of microphone arrays for improved speech recognition. *Proceedings of Eurospeech*, 2001.

[22] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, 3^{rd} edition, 1996.

[23] Sven Nordholm, Ingvar Claesson, and Nedelko Gribc. *Microphone Arrays: Signal Processing Techniques and Applications*, chapter 14. Optimal and Adaptive Microphone Arrays for Speech Input in Automobiles. Springer-Verlag, 2001.

Vita

Phil Townsend was born January 6, 1985 at Fort Campbell, KY. He completed his BSEE with honors at the University of Kentucky in 2007 Magna Cum Laude with minors in Mathematics and Computer Science on the National Merit Scholarship. Phil has previous technical experience as a co-op engineer in the Computer-Aided Design department at Cypress Semiconductor, also in Lexington, and has had work published through the proceedings of the 2009 IEEE SoutheastCon where he presented work on microphone array geometry.