A Reduced Complexity Adaptive Filtering System for Directional Listening

by

Michael Brian Goertz

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

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2001

Author .....
Department of Electrical Engineering and Computer Science
May 23, 2001

Certified by .....
Nathaniel I. Durlach
Senior Research Scientist
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Reduced Complexity Adaptive Filtering System for Directional Listening

by

Michael Brian Goertz

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2001, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

In this thesis, a method for reducing the computational complexity of an adaptive beamformer is explored. Fully adaptive systems can be computationally intensive and are not always practical for real-time systems. Fixed filter systems are robust in noisy environments and are simple to implement, however they do not achieve the maximal directionality that adaptive systems reach. By combining an adaptive beamformer with a traditional fixed filter beamformer, the strengths of each can be leveraged. The overall hybrid system requires less computation and is more robust to reverberation while approaching the performance of the fully adaptive beamformer. Preliminary simulation results show promise for such hybrid systems, showing both high gain at reduced complexity and robustness in reverberant environments not seen in adaptive systems alone.

Thesis Supervisor: Nathaniel I. Durlach
Title: Senior Research Scientist

Acknowledgments

First off, I would like to thank Joseph Desloge for all his time and energy through this whole process. The work that I have done is the direct result of his mentoring. His tireless editing of this document helped me organize the words on the page.

I want to thank Nathaniel Durlach for letting me take on the project one year ago. It has been a great learning experience for me, and I am happy to have had the opportunity.

My parents should not go without thanks. Without them I would not have had the chance to come to the great institution. I cannot thank them enough for opening that door.

And of course, thank you to all my friends, especially Emily, who supported me throughout this process. It was stressful at times and they kept me heading the right direction.

Contents

1 Introduction **9**

2 Background **11**
2.1 Fundamentals 12
2.2 Arrays 14
2.2.1 Directional Characteristics 15
2.3 Processing 17
2.3.1 Fixed Filtering 17
2.3.2 Adaptive Filtering 18
2.3.3 Comparison 19

3 Theory **21**
3.1 Structure of the Hybrid 21
3.2 Virtual Array Design 22
3.2.1 Virtual Array Design Metrics 24
3.2.2 Optimization of Virtual Array Elements 26
3.3 Virtual Array Design Considerations 27

4 Simulations **28**
4.1 Arrays 28
4.1.1 Real Array Configurations 29
4.1.2 Virtual Arrays 31
4.2 Acoustic Environments 34
4.2.1 Anechoic 34
4.2.2 Reverberant 34
4.2.3 Sources 35
4.3 Processing 35

5 Results **38**
5.1 Hybrid Performance Metrics 38
5.2 Results: Anechoic Environment 39
5.2.1 ULAs 39
5.2.2 Non-ULA arrays 41
5.3 Results: Reverberant Environment 43
5.3.1 ULAs 43

5.3.2 Non-ULA Arrays 44

6 Discussion and Conclusion **47**
6.1 Hybrid Performance Observations 47
6.1.1 Good Performance with low M' 47
6.1.2 Reduced Complexity 48
6.1.3 Increased Robustness 48
6.2 Future Work 48
6.2.1 Make the Virtual Elements More Similar 48
6.2.2 Develop an Optimized Hybrid System 49
6.2.3 Additional Research Ideas 51
6.3 Conclusions 51

A Virtual Array Figures **52**

B Simulation Details **61**
B.1 Environment Configuration 61
B.2 Sources 62

C Results **64**

List of Figures

2-1 Example of a basic beamforming system. Elements placed d = 20cm apart and a listening frequency of 825Hz. 12
2-2 The coordinate system used in this thesis. 13
2-3 Example of end-fire and broad-side sources to a ULA. 15
2-4 Directivity patterns of a simple ULA with an averaging filter at different frequencies. 16
2-5 A comparison of the relative computational complexity of adaptive and fixed filter processing. 20

3-1 Hybrid array processing system. 22
3-2 How a 6-element ULA can be transformed into a 3-element virtual ULA. 24

4-1 Circular arrays, inner diameter 15cm, outer diameter 20cm. 30
4-2 The circular random array used for testing. (a) top-down view of the array, x vs. y; (b) and (c) side-views, x vs. z and y vs. z. 30
4-3 Gain patterns for the 16 element virtual array at 1000Hz with the same dimensions as the real array it was constructed from. 32
4-4 Phase plots for the 16 element virtual array at 1000Hz with the same dimensions as the real array it was constructed from. 33
4-5 The naming convention used for the simulations 36

5-1 The average gain for each ULA based hybrid and the fully adaptive system versus the number of virtual elements in an anechoic environment. 40
5-2 Plots of gain versus the number of elements for four frequency bands for the ULA in an anechoic environment with a TJR_{input} of -10dB. . 40
5-3 The average gain for the non-ULA hybrid systems versus the number of virtual elements in an anechoic environment. 42
5-4 Plots of gain versus the number of elements for four frequency bands for the non-ULA arrays in an anechoic environment with a TJR_{input} of -10dB. 43
5-5 The average gain for each ULA hybrid system versus the number of virtual elements in a reverberant environment. 44
5-6 Plots of gain versus the number of elements for four frequency bands for the ULA in a reverberant environment with a TJR_{input} = -10dB. 45
5-7 The average gain for each generalized hybrid system versus the number of virtual elements in a reverberant environment. 46

5-8 Plots of gain versus the number of elements for four frequency bands for the non-ULA arrays in a reverberant environment. 46

6-1 One possible optimal segmentation of a hybrid system based on the fixed circular hybrid 50

A-1 Gain patterns for the 16 element virtual array based on the ULA at 1000Hz with variable virtual element spacing. 53
A-2 Phase plots for the 16 element virtual array based on the ULA at 1000Hz with variable virtual element spacing. 54
A-3 Gain patterns for the 16 element virtual array based on the circular array at 1000Hz with fixed virtual element spacing. 55
A-4 Phase plots for the 16 element virtual array based on the circular array at 1000Hz with fixed virtual element spacing. 56
A-5 Gain patterns for the 16 element virtual array based on the circular array at 1000Hz with variable virtual element spacing. 57
A-6 Phase plots for the 16 element virtual array based on the circular array at 1000Hz with variable virtual element spacing. 58
A-7 Gain patterns for the 16 element virtual array based on the randomized array at 1000Hz with variable virtual element spacing. 59
A-8 Phase plots for the 16 element virtual array based on the randomized array at 1000Hz with variable virtual element spacing. 60

B-1 Plot showing the source locations used for each of the ten iterations. Target Cell centered around 0 degrees. 63

C-1 ULA performance vs. frequency in an anechoic environment TJR input -20dB. 65
C-2 ULA performance vs. frequency in an anechoic environment TJR input -10dB. 65
C-3 ULA performance vs. frequency in an anechoic environment TJR input 0dB. 66
C-4 ULA performance vs. frequency in an reverberant environment TJR input -20dB. 66
C-5 ULA performance vs. frequency in an reverberant environment TJR input -10dB. 67
C-6 ULA performance vs. frequency in an reverberant environment TJR input 0dB. 67
C-7 Generalized array performance vs. frequency in an anechoic environment TJR input -20dB. 68
C-8 Generalized array performance vs. frequency in an anechoic environment TJR input -10dB. 68
C-9 Generalized performance vs. frequency in an anechoic environment TJR input 0dB. 69
C-10 Generalized array performance vs. frequency in an reverberant environment TJR input -20dB. 69

C-11 Generalized array performance vs. frequency in an reverberant environment TJR input -10dB. 70
C-12 Generalized array performance vs. frequency in an reverberant environment TJR input 0dB. 70

Chapter 1

Introduction

Directional listening is a way in which spatially separated audio sources can be isolated from one another and presented to a listener in a more intelligible manner. A listening environment often presents several audio sources, one of which is desirable to hear and the others of which are seen as jammers. The goal is to separate the target from the jammers so that the target is more clearly understood. The listener may be a human subject: e.g., a hearing impaired person wishing to isolate one conversation in a sea of noise at a cocktail party, or a soldier in the field wishing to hear the officer next to him from within a background of machinery and gun fire. The listener might also be a machine doing speech recognition in an automobile or intelligent room environment, where the target is submerged in road or room noise. In any case, the directional system needs to perform such that the target is amplified relative to the unwanted noise.

Acoustic directionality can be achieved through a number of different processes. One approach would be to use a highly directional single-element system, such as a parabolic dish. An alternative approach would be a system consisting of an array of acoustic elements (e.g., microphones) and a processor to filter the audio streams. Such a system can be made directional by choosing appropriate filters. This array processing system is of particular interest since it is flexible and allows for a wide range of processing techniques with varying degrees of performance.

Array processing can either be time-invariant or time-varying. Fixed, time-invariant

filtering has the benefit of simple processing, but since it is a fixed filter and aims to reduce the overall jammer gain, systems that actively cancel jammers can achieve higher gain. Such drawbacks limit the usefulness of such a system. A time-varying adaptive system, on the other hand, addresses some of these issues and in many cases produces better target isolation; however, this comes at the cost of increased computational requirements. To meet the goal of real-time processing, or at least minimizing computation for other goals (i.e., power consumption, using extra computational cycles for other tasks, etc.), a system that combines the benefits of both fixed and adaptive filter systems is attractive.

The result of this combination is a hybrid system that combines aspects of both fixed and adaptive systems, taking advantage of the non-overlapping strengths of each. This thesis presents such a system and explores one method for combining such systems and optimizing the performance over a sampling of implementations.

Chapter 2

Background

Array processing is a technique used to directionalize the output of a set of omni-directional receivers, or the transmission from omni-directional transducers. The variety of systems that can use beamforming are limitless and the types of signals involved can include audible and ultrasonic sound, as well as a various type of electromagnetic radiation. This research considers systems in which the elements are microphones. By creating a spatial array out of the microphones, spatially separated sources appear differently across the elements. Array processing works by exploiting these differences to cancel out some sources while maintaining others. This results in the creation of a beamforming system: a system that listens within the beam and mutes outside the beam.

Several factors determine how well such a system can perform and cancel unwanted sources. Firstly, the physical configuration of the elements plays a large role in the output. The number and spacing of the elements determine the maximum gain and the width of the beam. Secondly, the way in which the output from each of the elements is processed greatly effects the system output as well. In all the but the simplest of arrays, the signals from each element need to be filtered before summing. There are two ways of generating the filters. One is fixed, time-invariant processing, and the other is adaptive, time-varying processing.

2.1 Fundamentals

The most basic beamforming starts with two omni-directional elements placed some distance, d, apart. These two elements individually receive signals in all directions with the same normalized gain of unity. Now if the signals from the two elements are summed together, and renormalized by dividing by two, the resulting system will have non-uniform directional characteristics. Figure 2-1 shows the example of two microphones separated by 20cm. When a source of frequency of 825Hz is broadside to the array, (i.e., perpendicular to the array axis), the gain is unity. When a source is endfire, (i.e., along the array axis), the gain is zero and the source is nulled out. This pattern arises from the fact that the element spacing of 20cm is half the wavelength, $\lambda/2$, of the 825Hz source. When the source is broadside, the received signals add constructively. When the source is endfire, the received signals are out of phase by 180 degrees and add destructively or subtract.

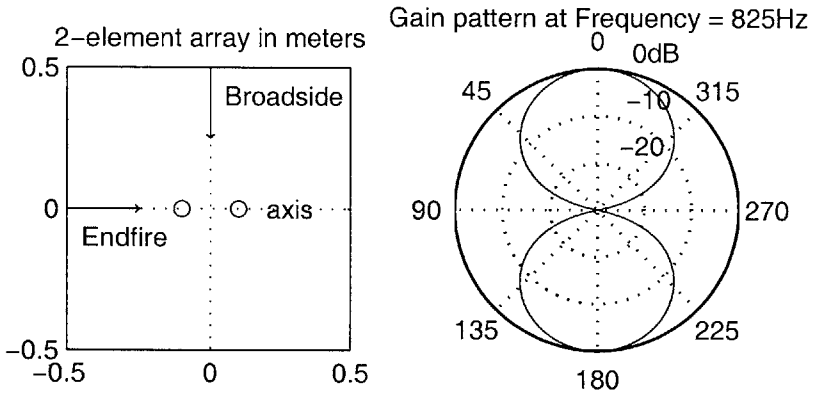

Figure 2-1: Example of a basic beamforming system. Elements placed $d = 20$cm apart and a listening frequency of 825Hz.

Differences in received signals at each of the array elements can be exploited to separate individual sources. This happens because each element has a source-to-element frequency-domain transfer function, $H_m(\rho, \theta, \phi, \omega)$, that is dependent on the element m and the source location ρ, θ, and ϕ. Figure 2-2 shows the source location coordinate system.

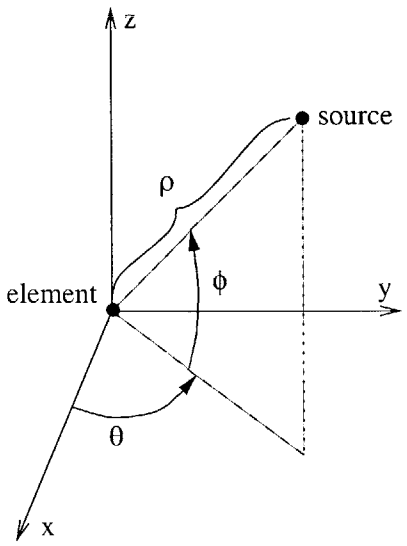

Figure 2-2: The coordinate system used in this thesis.

Combining the source-to-element transfer functions yields a vector of the following form,

$$\overline{H}_M(\theta, \phi, \omega) = \left[\begin{array}{cccc} H_1(\theta, \phi, \omega) & \ldots & H_M(\theta, \phi, \omega) \end{array} \right]^T. \tag{2.1}$$

In an anechoic environment with omni-directional microphones, under the assumption of far-field sources, then the distance to each element, ρ, will have little relative effect on each impulse response and it can be suppressed. The location of a source will then dictate the relative delay seen by each receiver, with little change in amplitude across the array [7].

Each discrete frequency bin can then be filtered independently using a complex weight vector, $\overline{W}_M(\omega)$. The output, $Y(\omega)$, is given by,

$$Y(\omega) = \overline{W}_M(\omega)^H \overline{X}(\omega),$$

where ^{H} is the complex conjugate transpose, $\overline{X}(\omega)$ is the array input signal, and the

complex weight vector, $\overline{W}_M(\omega)$, is defined as,

$$\overline{W}_M(\omega) = \begin{bmatrix} W_1(\omega) & \dots & W_M(\omega) \end{bmatrix}^T. \tag{2.2}$$

Each element of $\overline{W}_M$ multiplies the signal from one element and the results are summed to form the output.

2.2 Arrays

An array is any configuration of two or more receiving elements. As stated above, the physical array configuration plays an important role in determining the behavior of the complete beamforming system. Certain types of arrays lend themselves better to certain performance characteristics. The simplest of all arrays is the Uniform Linear Array (ULA). In this type of array, the elements are equally spaced along a straight line, as illustrated in Figure 2-3. This array serves as a basis for describing array performance, with more generalized array configurations being introduced later.

Ideally, an array would have as large a span with as high an element density as possible. This would ensure the greatest beamforming ability. However, most practical systems are limited by space, thus limiting the span, and limited in processing power and cost, thus limiting the maximum number of elements. In order to chose an optimal array given realistic constraints, the way in which span and density effect the performance must be understood. The interelement spacing greatly effects the array's performance at high frequencies, and the span of the array effects the performance at low frequencies. As a basis, the desired spacing should be no more than half a wavelength, λ, at the highest frequency, $d = \lambda/2$, and the ULA length should be at least $\frac{\lambda}{\theta_{sector}}$ where θ_{sector} is the width of the sector, as specified in [4]. These dimensions make sense since: at low frequencies, the array needs to be large to spatially capture the slowly varying waveform; at high frequencies, the elements must be close to enough to spatially sample the waveform without spatial aliasing and retain all of the quickly varying information. These guidelines help to maximize

the differences perceived by each microphone and thus maximize the performance.

To further complicate the array design, most acoustic signals of interest are broadband, and the dimensions of the array must be considered over the entire frequency range. To meet the guidelines just outlined, certain considerations need to made for very low and very high frequencies. As frequency approaches zero, λ approaches infinity. This means that the array span would also become very large. Alternately, as frequency increases, λ can become very small as does the interelement spacing. So over a large frequency range, the span must be large and the microphones must be close together. For a bandwidth of 300Hz-15kHz and a beam width of 22.5 degrees, 100 microphones would be needed spanning more than 1 meter. This is obviously not practical for many applications. Based on the system requirements, a compromise needs to be made. The types of filters and the type of array used can have a large impact on the performance of the array as well and can help determine the best trade-off.

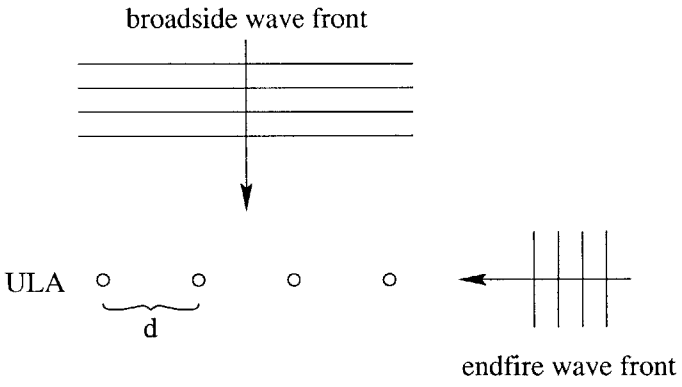

Figure 2-3: Example of end-fire and broad-side sources to a ULA.

2.2.1 Directional Characteristics

It is not sufficient to merely introduce guidelines by which an array should be configured. An understanding is needed of how the directionality of an array varies across frequency, along with tools to help quantify its performance.

At different frequencies, the same array can exhibit varying reception patterns.

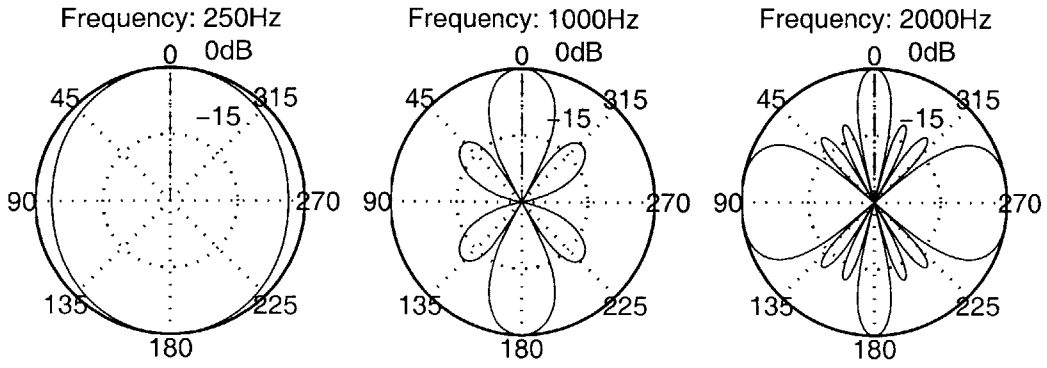

Figure 2-4: Directivity patterns of a simple ULA with an averaging filter at different frequencies.

Figure 2-4 shows the reception pattern evaluated at three frequencies for a 4-element ULA aligned broadside to 0 degrees with interelement spacing $d = \lambda/2 = 16.5$cm at 1000Hz using the averaging filter, $\overline{W} = \frac{1}{4}[1, 1, 1, 1]^T$. When the array is evaluated at 1000Hz, the output exhibits good directionality with a narrow beam. At 250Hz, the array barely has a beam at all, and is nearly omni-directional. At 2000Hz, when the interelement spacing is λ, the array has a very narrow beam in the target direction. However, there is a much larger beam 90 degrees off axis. At this frequency, the array inter-element spacing is large enough to cause spatial aliasing, sources coming from the endfire direction exhibit the same phase as a source in the broadside direction. In each of the figures, the array exhibits a front/back symmetry. This is a result of the physical symmetry of the array about its axis.

As can be seen, the variation in reception patterns makes it desirable to have some simple way of quantifying how directional they are. Fortunately, such a metric already exists. It is the directivity index [10] and is calculated from the following:

$$D = \frac{\text{Target Location Output Power}}{\text{Output Power Averaged Over All Locations}} \tag{2.3}$$

$$= \frac{W^H(\omega)H(\theta_{target}, \phi_{target}, \omega)H^H(\theta_{target}, \phi_{target}, \omega)W(\omega)}{W^H(\omega)S_{xx}W(\omega)}, \tag{2.4}$$

where S_{xx} is the cross-spectral density matrix, given by,

$$\frac{1}{4\pi}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2}\overline{H}_M(\theta,\phi,\omega)\overline{H}_M^H(\theta,\phi,\omega)sin(\theta)\,d\theta\,d\phi. \tag{2.5}$$

2.3 Processing

There are two distinct types of filtering that can be done. Each has its strengths and weaknesses and is well suited to particular situations. Time-invariant filtering, as the name implies, has a set of fixed filters that do not change with time. Time-dependent processing varies the filter based on a set of criteria derived from the environment.

2.3.1 Fixed Filtering

Fixed filter systems work by implementing a set of filters that will minimize the output power from all potential non-target locations. These filters are created prior to run-time and thus can be implemented as simple FIR filter banks, with minimal computational complexity.

The filters for such systems are typically chosen to maximize the directivity index as given in Equation 2.4. There is a closed form solution to the maximization given by [1, 5],

$$W^H(\omega) = (H^H(\theta_{target},\phi_{target},\omega)S_{xx}^{-1}(\omega)H)^{-1}H^H(\theta_{target},\phi_{target},\omega)S_{xx}^{-1}(\omega). \tag{2.6}$$

In certain cases, when the array is small compared to λ, the magnitude of the weight vector, $||\overline{W}||$, can be very large. Uncorrelated noise and filter input, such as wind noise or quantization error, will be multiplied by the magnitude of these weights and could have a large negative impact on the output of the system. To eliminate this problem, [10] suggests a method of iteratively finding the optimum weights by adding a certain amount of white noise to S_{xx} and then solving Equation 2.6.

2.3.2 Adaptive Filtering

Adaptive, time-variant beamforming systems operate by using a method of feed back to create a set of filters at run-time. These systems aim to actively cancel noise sources by doing a constrained minimization of the system output power. By actively cancelling noise sources, these systems can achieve much higher directional gain, especially in the presence of strong noise, than traditional time-invariant designs.

Generally, the adaptive system uses linear constraints to maintain a gain of unity in the target direction. The work of [11] presents a Linearly Constrained Minimum Variance Beamformer (LCMV). In this system, the filters are found with the following minimization:

$$\begin{aligned} \min_{W} \mathrm{E}[|Y|^2] &= \{W^H R_{xx} W\} \text{ such that } C^H W = f, \\ \text{where, } X &= \text{array data} \\ C &= \text{constraint matrix} \\ f &= \text{response vector} \\ R_{xx} &= E\{xx^H\} \\ \text{output: } Y &= W^H X. \end{aligned}$$

The optimum weights can be found with the following:

$$W = R_{xx}^{-1} C [C^H R_{xx}^{-1} C]^{-1} f.$$

An example constraint matrix C along with f can be chosen such that the target location is preserved with unit gain: $f = 1$ and $C = H_{target}$. Adaptive processing can often result in higher jammer cancellation over the fixed filter systems. All of this comes with a trade-off, however. Such systems have substantial computational requirements. The solution for the optimum weights requires several matrix inversions as well as matrix-by-matrix multiplies, which make this system much more computa-

tionally intensive than the fixed filter system. Also, there is the possibility of target cancellation. This can happen when the target is not exactly in the expected direction or in a reverberant environment when there are signals that sound like the target coming from non-target directions. So the robustness of adaptive systems is an issue that needs to be addressed.

Several different types of adaptive filtering systems exist. This thesis will focus on one such system in developing the hybrid system. The system of interest is the LENS (Location-Estimating, Null-Steering) algorithm developed in [3]. This method is based on the LCMV method describe above. Specifically, it considers the LCMV problem with the single constraint of preserving the target location. This is achieved by setting $C = H(\theta_{target}, \phi_{target})$ and $f = 1$. LENS then parameterizes this problem by converting it from a minimization over the elements of W to a minimization over the LENS parameter set β, where W can be expressed as $W = W(\beta)$. The LENS parameter set β is chosen so that each parameter β_i corresponds to one null location in the adaptive null-steering process. By observing these parameters, LENS can detect and prevent inadvertent target-signal cancellation. This makes LENS a more robust adaptive beamforming technique.

It should be noted that LENS is intended to operate on ULA configurations. It can be generalized to other configurations, however, through the use of pre-processing designed to make the general array appear ULA-like near the target location.

2.3.3 Comparison

Adaptive filtering brings with it the ability to cancel out strong jammers as they appear and change within the environment. However, this comes at the cost of complex computation. Alternately, fixed filtering is be simpler to implement and more robust to target cancellation, but realizes less overall target gain. The end application requirements ultimately determine the appropriate choice of processing.

It is obvious that a problem exists with the complexity of the adaptive system. Figure 2-5 compares the computational complexity of fixed and adaptive systems. Given M microphones, fixed array processing systems generally require $O(M)$ com-

putations while adaptive systems generally require $O(M^2)$-$O(M^3)$ computations. As shown, the number of adaptive-system computations grows rapidly with increasing M.

Given the strengths and weaknesses of both adaptive and fixed filter processing, there is good reason to believe that these two types of systems could be combined into a hybrid system that maintains a level of performance approaching that of a fully adaptive system yet achieves a computational complexity and robustness similar to that of fixed systems. Work has been done to combine these two systems with good results, but in a non-general way [8]. In the coming chapters, one method of obtaining a general hybrid system is outlined and demonstrated.

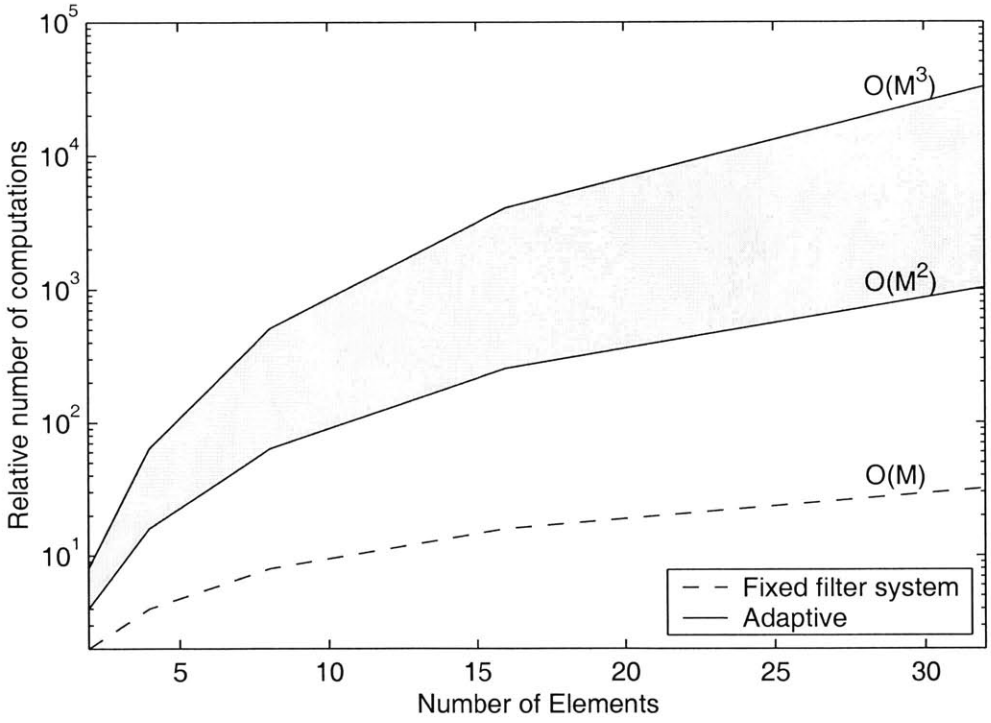

Figure 2-5: A comparison of the relative computational complexity of adaptive and fixed filter processing.

Chapter 3

Theory

As discussed in Chapter 2, the fixed and adaptive filter approaches to the array processing problem each have strengths and weaknesses. Ideally, a hybrid system would combine fixed and adaptive techniques to produce a system that builds on the strengths of both: it would exhibit superior target enhancement as with the adaptive systems, but it would do so robustly and with less complexity similar to fixed systems.

3.1 Structure of the Hybrid

The hybrid approach developed in this research combines fixed filter pre-processing with adaptive post-processing, as shown in Figure 3-1. The fixed filter stage transforms the M array inputs into a virtual array of M' highly-directional elements. The adaptive stage then processes the data from the virtual array. When $M' < M$ the adaptive processor computational requirements decrease, as shown in Figure 2-5.

This system can maintain performance even for $M' < M$ since at certain frequencies, the non-adaptive virtual array elements can be very directional. As shown in Figure 2-4, M' is adjusted according to the directionality of the virtual elements: M' is smaller for more directional virtual elements and M' is larger (up to $M' = M$) for less directional virtual elements. Furthermore, depending on the end array configuration, it may be possible to build a hybrid fixed-adaptive system that is more robust in reverberant environments, or that enhances directionality relative to the

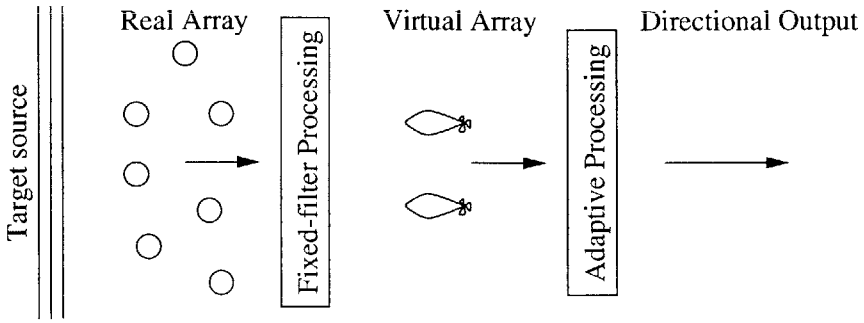

Figure 3-1: Hybrid array processing system.

fully adaptive system.

Note that all new design issues in this hybrid system arise only in the creation of the virtual array. Because the virtual array is designed to mimic the properties of real arrays, no modifications to the adaptive processing are required; the adaptive system operates directly on the virtual array elements. For this reason, this chapter is devoted to the design of the virtual array.

3.2 Virtual Array Design

The idea of a virtual array is rather simple: it should output signals from each virtual element that behave, from a signal analysis point of view, the same way real directional elements would behave. The virtual array is constructed and evaluated by how closely a few characteristics mimic an idealized array. These characteristics are:

- **Phase:** The phase between signals from each of the virtual elements should behave in a fashion that is useful to the adaptive array processor. In this case, the signals need to behave like those coming from the elements of an idealized ULA.
- **Gain:** The gain of each element should be highly directional towards the target sector.

Using these characteristics, a virtual array can be created by performing a constrained maximization of the gain while keeping desired phase properties.

Virtual Arrays: A simple example

To illustrate the construction of a simple virtual array, consider the task of transforming a six-element ULA into a three-element ULA ($M = 6$ and $M' = 3$), as pictured in Figure 3-2. This figure shows that this can be done by creating three four-element sub arrays from the six elements, and applying the averaging filter, $\overline{W} = \frac{1}{4}[1,1,1,1]^T$ to each sub-array. The resulting weight matrix would be

$$\overline{W}_{virt} = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \end{bmatrix},$$

where each column of $\overline{W}_{virt}$ is a spatial filter that transforms the original array inputs into one virtual element input. More specifically, virtual element 1 would be the average of real elements 1 through 4, virtual element 2 would be the average of real elements 2 through 5, and finally, virtual element 3 would be the average of elements 3 through 6. The new virtual elements would each exhibit the directional characteristics shown in Figure 2-4, as well as have an interelement phase relationship equal to that of three real elements with a spacing d, as shown in Figure 3-2.

The limitation of this technique for creating virtual arrays is that it only applies to ULA input arrays and ULA virtual arrays. In the remainder of this chapter, the construction of virtual arrays is generalized to arbitrary real and virtual array configurations.

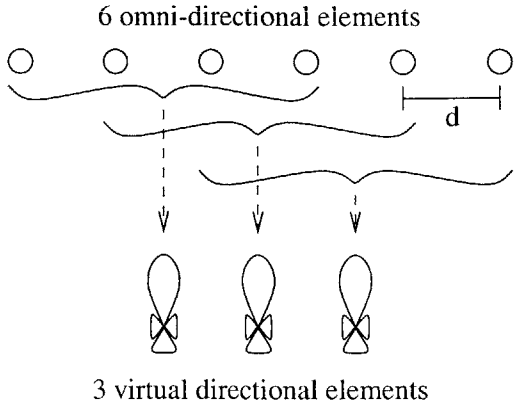

Figure 3-2: How a 6-element ULA can be transformed into a 3-element virtual ULA.

3.2.1 Virtual Array Design Metrics

The basic virtual array idea is to transform M real-element signals into M' virtual-element signals, much like the example outlined above. The problem with the preceding example is that it requires both the real array and the virtual array to be ULAs with element spacing d. Ideally, we would like a general approach that would enable one to transform any real array configuration into any virtual array configuration.

The following sections describe the characteristics that are required to generate generalized virtual arrays. The directivity index introduced in Equation 2.4 is modified to reflect the idea that a target region may be more desirable than a single target location. Also important is a notion of how the virtual array compares to a desired array configuration. To address that issue, the inter-signal phase of the virtual elements will be measured to help ensure that the virtual array can be made to act like a real array. Finally, a measurement of the noise sensitivity is needed so that the effect of of uncorrelated noise on the system performance can be limited.

Directivity

Directivity as defined in Section 2.2.1, is inadequate for describing the performance of certain spatial-filtering applications because it assumes a single target direction. More realistically, possible target locations may span an entire spatial sector. The

directivity index needs to be modified such that it reflects the cell enhancement and not just a single-location enhancement. This may be accomplished by defining

$$D_{sector}(\omega) = \frac{\text{Average output over the sector}}{\text{Average output power over all locations}} \tag{3.1}$$

$$= \frac{W^H(\omega)S_{cc}(\omega)W(\omega)}{W^H(\omega)S_{xx}(\omega)W(\omega)}, \tag{3.2}$$

where $S_{cc} = \int_{sector} HH^H \, d\theta$ is the covariance matrix within the cell and S_{xx} is the isotropic noise for the array as defined in Equation 2.5.

The average sector gain should be unity, but this alone is not enough to guarantee satisfactory filters. For large sectors, it is possible to choose a filter with large gain in part of the sector and low gain in another part, which is obviously undesirable. Thus, an error variance metric will be associated with the sector directivity index. Use of the error variance parameter ensures that the gain at each point in the sector will, with some error, be close to unity.

$$DV_{sector} = \frac{1}{\theta_{sector}} \int_{sector} [|W^H H| - 1]^2 \, d\theta,$$

where θ_{sector} is the width of the sector.

Interelement Phase

A key issue in building a virtual array is characterizing the interelement phase relationships. These relationships must closely resemble those of an idealized array. Due to the other constraints on the system, the interelement phase will only be considered within the sector. Phase outside of the sector can be ignored since the adaptive system is primarily concerned with the "in-sector" phase in order to prevent target cancellation.

The phase of the system is constrained by defining a desired-phase characteristic over the target sector and then setting a maximum variance on the difference between the actual virtual array phase and the desired phase. Desired phase is computed by the following method: first, the virtual array is centered at the center of the real

array. Then, each virtual element is calculated and constrained to have a phase in the sector equal to that of a real element at the desired location. P_{sector} describes how the phase variation is measured,

$$P_{sector} = \frac{1}{\theta_{sector}} \int_{sector} |\angle H_{virt}(\theta, \omega) - \angle H_{desired}(\theta, \omega)|^2 \, d\theta,$$

$H_{desired}$ is the source-to-element transfer function of the ideal virtual element, θ_{sector} is the width of the sector, and $H_{virt} = W_M^H H$.

Noise Sensitivity

Limiting the noise sensitivity limits the maximum gain of uncorrelated noise at the inputs of the system. This limits the negative effect of such factors as wind noise, quantization error, random placement errors of the array elements, and any other sources of uncorrelated random noise at the input. The noise sensitivity is defined as,

$$NS = W_M^H(\omega) W_M(\omega).$$

The filters for each virtual element are limited individually.

3.2.2 Optimization of Virtual Array Elements

Using the defined parameters, a set of virtual elements can now be created by optimizing the weights given a set of constraints on those parameters. Each virtual element is solved for individually using the following method:

- Maximize D_{sector}
- Constrain NS, DV_{sector} and P_{sector} to some maximum value, so that the virtual array closely matches the desired array.

$$\overline{W}_{virt} = \arg\max_{\overline{W}_{virt}}(D_{sector}) \text{ such that } DV < threshold$$
$$P_{sector} < threshold$$
$$NS < threshold.$$

The constraints need to be chosen small enough such that the virtual array performs like the desired array through the sector. However, setting them too small will limit the maximum directivity and possibly make it impossible for the numerical optimization procedure to find a solution.

3.3 Virtual Array Design Considerations

One key issue that needs to be addressed is the spacing of the virtual elements. Since all virtual arrays being considered are ULAs, the guidelines stated in Section 2.2 will be considered: the desired spacing should be no more than $\lambda/2$ and the ULA length should be at least $\frac{\lambda}{\theta_{sector}}$ where θ_{sector} is the width of the sector, as specified in [4] for optimal directivity. For a real array, it is desirable to meet these criteria, but the effect on virtual array performance is unknown and will be evaluated.

When designing the virtual arrays, the interelement spacing will be chosen at design time. The complexity added to the problem by trying to optimize over spacing as well as over the other parameters would be tremendous. The spacing that is chosen will be dependent on the real array that the virtual array is derived from as well as the wavelength being considered. Care must be taken to ensure that virtual elements are not placed outside of the real array: these add no additional spatial information and only serve to limit the ability of the numerical optimization to find a usable solution.

Chapter 4

Simulations

This chapter presents several hybrid array processing systems and simulations of the performance obtained with them. Three real arrays were tested within two different acoustic environments. The test systems evaluated were chosen to demonstrate the expected performance of this type of hybrid spatial filter. Each test configuration varied both the number of virtual array elements, M', and the virtual array parameters. By varying M', the performance change associated with less adaptive filtering was observed.

Each real array was transformed into a virtual array according to the details outlined in chapter 3. The virtual arrays were designed to work directly with the LENS adaptive processing system.

4.1 Arrays

The real array configurations simulated those used in real applications. The virtual arrays were designed to interface with the LENS processor and thus were all ULAs. Each array chosen was used to demonstrate a particular feature of the hybrid system with the hopes of better understanding the system so that it can be designed to work in more complex environments.

4.1.1 Real Array Configurations

There are three real array configurations that were used for testing: the ULA, the circular array, and the randomized array. The details of each is presented below.

ULAs

Uniform Linear Arrays are not usually desirable for general applications because of their axial symmetry. However, their simplicity and well understood operation makes them good for demonstration, especially since they can be directly processed with the LENS processing system. This feature makes them ideal for comparison with any virtual arrays created. The dimensions of the real ULAs used are typical of general applications. The interelement spacing was 2.75cm, or $\lambda/2$ at 6kHz. Tests were done on ULAs with M = 2, 4, 8 and 16 elements. As M varied, the element spacing was held fixed, to maintain good performance at high frequencies by avoiding spatial aliasing, and the array span was varied. At lower frequencies, the maximum directionality is decreased due to the smaller span.

Circular Arrays

Circular arrays were also considered for study since they are likely candidates for a head or helmet-mounted arrays. They exhibit rotational symmetry, which has the advantage of allowing a single set of filters to be used to spatially filter target signals from several possible locations.

The particular M-element circular arrays chosen consisted of a pair of concentric circles, each with $M/2$ elements (Figure 4-1). The inner circle is rotationally offset from the outer circle, giving the array a slightly more interesting spatial diversity. While the exact circular array configuration is somewhat arbitrary, it is not an unrealistic array and it has some historical importance within the research group as a configuration used in the past.

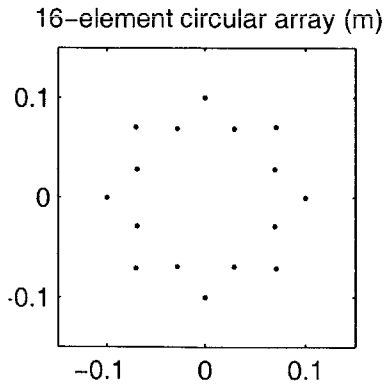

Figure 4-1: Circular arrays, inner diameter 15cm, outer diameter 20cm.

Randomized Arrays

The random array chosen for study was the 16-element circular array described above with each element perturbed a uniform random amount in 3-space ± 1cm. This represents a realistic scenario where a circular array is in the field and poor manufacturing and rough handling have caused the elements to be moved from their original location. Calibration of the array would need to take place in order to measure the new microphone locations. Figure 4-2 shows the array used in the tests performed.

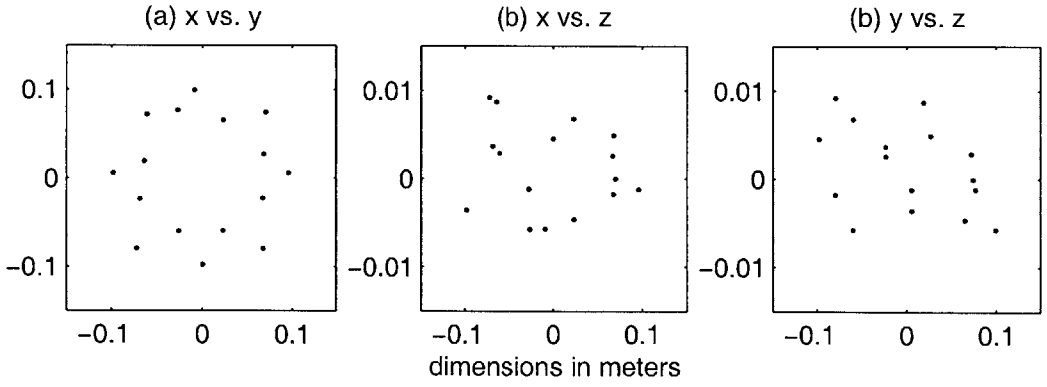

Figure 4-2: The circular random array used for testing. (a) top-down view of the array, x vs. y; (b) and (c) side-views, x vs. z and y vs. z.

4.1.2 Virtual Arrays

All virtual arrays were ULAs since the LENS system has been developed only for this case. The details of these virtual arrays were varied to understand the best approach to the virtual ULA design problem. Several virtual array configurations are considered:

- Virtual arrays built from real ULAs
- Virtual arrays built from real circular arrays
- Virtual arrays built from real randomized arrays

Within each of the real array configurations, the parameters of the virtual array configuration were varied to create different hybrids.

Real ULAs

For the real ULA configurations, the virtual compressed array presented in Section 3.2 was tested as a baseline virtual array because its attributes are easily understood. Specifically, a 16-element real ULA was transformed into an 8, 4 and 2 element compressed virtual ULA. For the 8-element case, the optimal maximum-directivity filter for a 9-element ULA was shifted across each of the 8 groups of elements. For the 4 and the 2 element configuration, the same 9-element optimal filter was used on the center 4 or 2 elements.

The ULA baseline was then compared to the general virtual array design method of Chapter 3. In order to see what additional performance might be gained by incorporating all 16 microphones of the real ULA into each virtual element, a virtual array with the same dimensions as the real array was generated.

As shown in Figures 4-3 and 4-4, this virtual array did not exhibit the same uniformity across elements (Note that only elements 1-8 out of 16 are shown, due to the symmetry for the real and virtual arrays; elements 16-9 look the same as elements 1-8). The gain through the azimuth plane is noticeably different between elements. The sector is marked by the dashed lines centered around 0 degrees. An even more

significant difference is that the phase is constrained within the sector only. Once again, the sector is noted by the dotted line centered about 0 degrees.

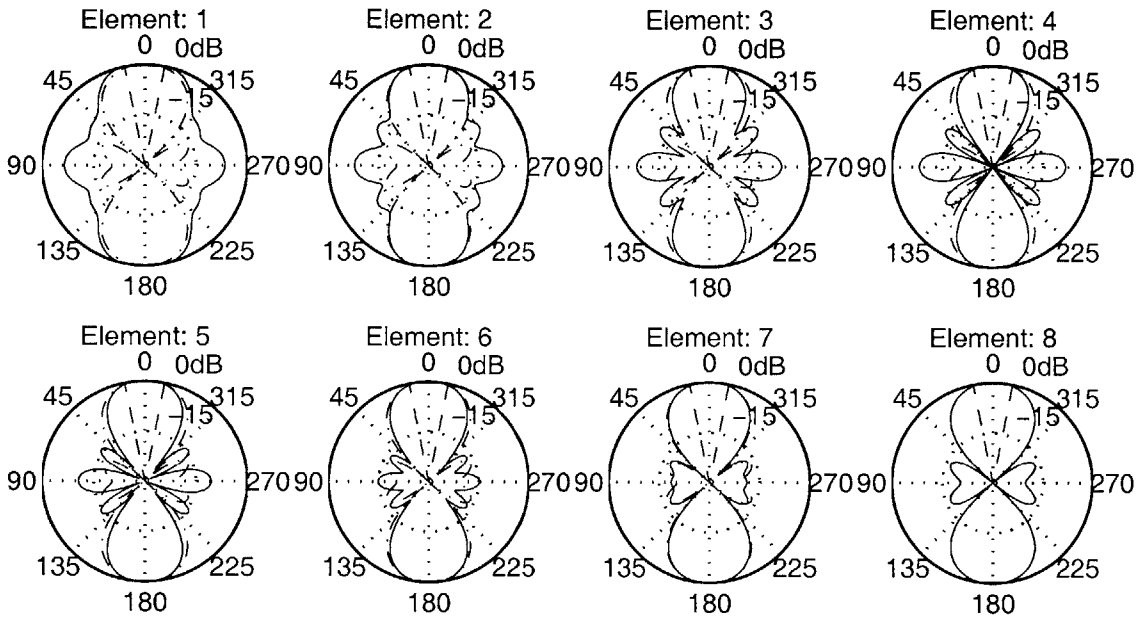

Figure 4-3: Gain patterns for the 16 element virtual array at 1000Hz with the same dimensions as the real array it was constructed from.

Finally, an array was generated from the real ULA that varied the spacing of the elements proportional to the wavelength. This was done in order to study the effect that different interelement spacing would have on the effectiveness of the adaptive processing. For this hybrid, the inter-element spacing was chosen to be the minimum of $\lambda/10$ or $\frac{\text{real array span}}{16}$.

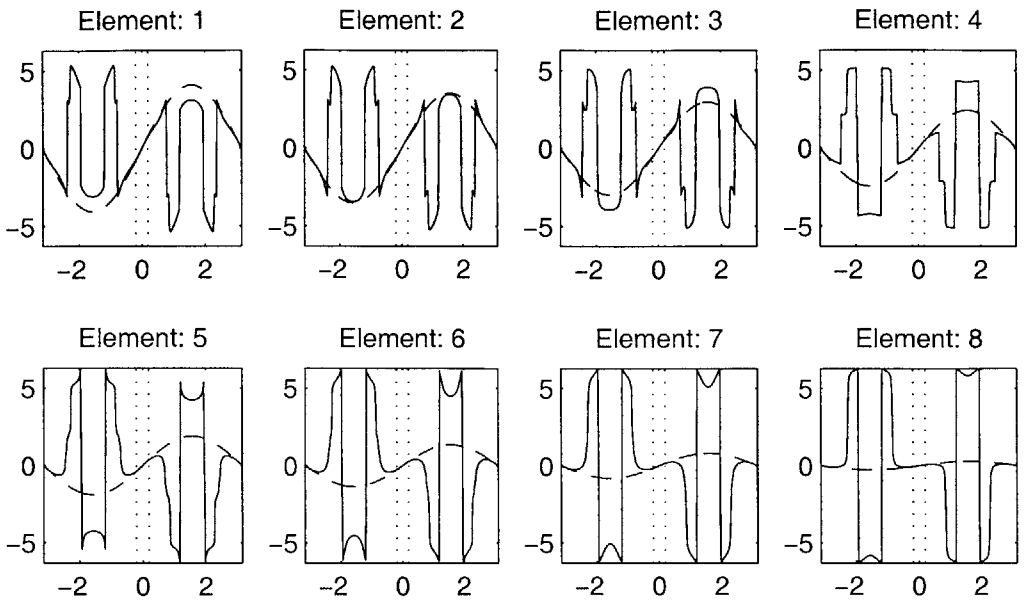

Figure 4-4: Phase plots for the 16 element virtual array at 1000Hz with the same dimensions as the real array it was constructed from.

Other Real Arrays

In addition to the real ULA, the circular and randomized arrays were also used to generate virtual arrays. The virtual array configuration was varied in much the same way, by using both fixed and variable virtual-element spacing.

For these arrays, the parameters controlling the gain, the phase variance, and noise sensitivity were held constant. While these can obviously effect the performance of the system, it was thought more important to study the effects of variations in the array configuration within the time constraints of this thesis. For hybrid systems incorporating virtual array sizes smaller than 16, the inner elements from the 16-element virtual ULA were used.

Since these arrays have no real counterpart that can be directly processed by LENS, the optimal maximum directivity fixed-filter beamformer was used as a benchmark system for comparison.

4.2 Acoustic Environments

All systems were tested using a consistent set of environments. Source location and power, and room characteristics were chosen to test different aspects of each systems performance.

4.2.1 Anechoic

The anechoic tests look at the performance of the system in the absence of acoustically reflective objects, i.e. in an empty infinite room. Anechoic tests are useful to understand the underlying performance of the hybrid without the complexity that a real environment adds. The results of these tests are not realistic for predicting performance in most environments. If the hybrid were used in an open field without many acoustically reflective objects, then these tests would be a good indicator of performance.

4.2.2 Reverberant

The reverberant tests are more useful for understanding the performance of the system in a room environment. In these situations, the system needs to be robust to avoid target cancellation. Reflections of the target from outside the sector can be filtered in such a fashion that the target will be cancelled while meeting the constraint of keeping the target gain at unity. Also, reverberated jammer images can be picked up by the array from within the sector. Finally, depending on the environment, reflections might number too many with high volume for the adaptive system to adequately cancel them.

The simulated room in these test was measured to have a direct-to-reverberant ratio of -0.56 dB. A fairly robust system was needed to avoid target cancellation and provide a useful reduction of jammer levels.

4.2.3 Sources

Each system was tested with 6 sources including the target source. In each case, the target source was placed in the center of the sector and the jammer locations were randomly chosen subject to the single constraint that they lie outside the sector. Each test was averaged over 10 sets of jammer locations. For consistency, each real array configuration was simulated with the same source locations. Appendix B contains the specific source configurations used.

The power of all of the jammer sources were equalized, summed and normalized to have unit energy. The target source power was then adjusted to meet a desired input Target-to-Jammer Power Ratio, TJR, requirements, where TJR is defined as,

$$TJR = \frac{P_{target}}{P_{jammers}}.$$

Each hybrid system was tested with TJR input levels of -20dB, -10dB and 0dB.

4.3 Processing

The virtual array filters were generated for a complete set of frequencies at a sampling rate of 11025Hz and a FFT block length of 256, for a total of 129 complex frequency bins per array. Certain parameters of the virtual array filters were based on the the frequency, since the wavelength has such a large effect on the performance of an array.

The test signals were white noise generated at runtime for each of the systems. The signal length was 5 seconds, 55126 samples, and the performance was evaluated from the last 2 seconds of the output signal. This gives the adaptive subsystem time to adapt to the jammer configuration.

The FFT based processing was done on 256 sample length blocks. The blocks overlapped by 128 samples and were windowed with $\frac{1}{2}(1 + cos[n])$.

The specific tests run are listed in Table 4.1. Shown are the number of real and virtual elements in each hybrid, environment characteristics, and the TJR of the input. Note that the real-array tests do not include a number for virtual elements

since the LENS system could directly process the real ULA input. Figure 4-5 shows specifically how each test was named. This naming convention is used for all included results.

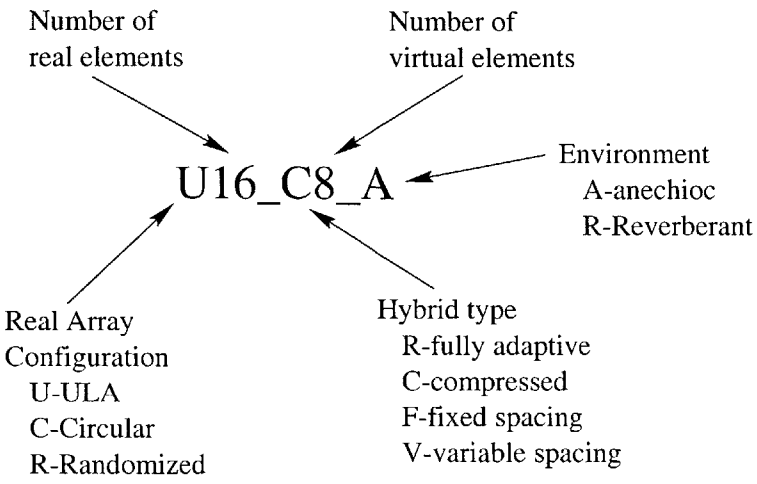

Figure 4-5: The naming convention used for the simulations

	Real Array	Name	Filter Type	M Real Elements	M' Virtual Elements (X)	Environment	Notes
Fully Adaptive	ULA	U16_R_A	N/A	16	N/A	Anechoic	TJR_{input} = -20, -10, and 0dB
		U16_R_A	N/A	8	N/A	Anechoic	TJR_{input} = -20, -10, and 0dB
		U4_R_A	N/A	4	N/A	Anechoic	TJR_{input} = -20, -10, and 0dB
		U2_R_A	N/A	2	N/A	Anechoic	TJR_{input} = -20, -10, and 0dB
		U16_R_R	N/A	16	N/A	Reverberant	TJR_{input} = -10dB
		U8_R_R	N/A	8	N/A	Reverberant	TJR_{input} = -10dB
		U8_R_R	N/A	4	N/A	Reverberant	TJR_{input} = -10dB
		U2_R_R	N/A	2	N/A	Reverberant	TJR_{input} = -10dB
Hybrid Systems	ULA	U16_CX_A	Compressed	16	8,4,2	Anechoic	TJR_{input} = -20, -10, and 0dB
		U16_VX_A	Variable	16	16,8,4,2	Anechoic	TJR_{input} = -20, -10, and 0dB
		U16_FX_A	Fixed	16	16,8,4,2	Anechoic	TJR_{input} = -20, -10, and 0dB
		U16_CX_R	Compressed	16	8,4,2	Reverberant	TJR_{input} = -20, -10, and 0dB
		U16_VX_R	Variable	16	16,8,4,2	Reverberant	TJR_{input} = -20, -10, and 0dB
		U16_FX_R	Fixed	16	16,8,4,2	Reverberant	TJR_{input} = -10dB
	Circular	C16_VX_A	Variable	16	16,8,4,2,1	Anechoic	TJR_{input} = -20, -10, and 0dB
		C16_FX_A	Fixed	16	16,8,4,2,1	Anechoic	TJR_{input} = -20, -10, and 0dB
		C16_VX_R	Variable	16	16,8,4,2,1	Reverberant	TJR_{input} = -20, -10, and 0dB
		C16_FX_R	Fixed	16	16,8,4,2,1	Reverberant	TJR_{input} = -20, -10, and 0dB
	Randomized	R16_VX_A	Variable	16	16,8,4,2,1	Anechoic	TJR_{input} = -20, -10, and 0dB
		R16_VX_R	Variable	16	16,8,4,2,1	Reverberant	TJR_{input} = -20, -10, and 0dB

Table 4.1: Listing of each of the tests performed.

Chapter 5

Results

Presented here are the results of the simulations listed in the previous chapter. The performance metrics used to analyze the simulations are outlined, and the results are given. The anechoic environment results are presented first, and then the reverberant environment results.

5.1 Hybrid Performance Metrics

It is important to have metrics that can be used to evaluate the performance of each hybrid. The Target-to-Jammer Ratio improvement and the computational complexity summarize the important aspects of each hybrid's performance and characterize its behavior.

Target-to-Jammer ratio (TJR) improvement is the metric used to analyze the performance of each of the simulations. By looking at the TJR of the output signal and comparing it with the TJR of the input, the gain of the system can be computed,

$$Gain = TJR_{improvement} = TJR_{output} - TJR_{input},$$

where the TJR is in dB.

The TJR improvement should not be considered alone however. It is important to

keep in mind that a system with a high input TJR needs less improvement to create usable output. Conversely, a system with a very low TJR input needs quite a bit of improvement in order to make intelligible output possible, although, a good output TJR does not ensure that the output is intelligible.

TJR improvement is considered in several ways. First, the average gain over all frequencies is computed. This provides a quick way to survey the performance of each setup. Second, for particular systems of interest, the overall average gain is broken down into four frequency bins roughly log spaced. Lastly, plots of the system gain versus frequency are available in Appendix C. These TJRs show how the various hybrid systems perform across frequency and suggest ways to combine different hybrid systems in different frequency bands.

Computational Complexity of each of the hybrid systems is directly related to the number of virtual elements that the adaptive subsystem must operate upon. This is a good approximation of the overall system computational complexity, since the adaptive system and in particular, the number of adaptive degrees of freedom, are the dominant influence, as seen in Figure 2-5. A complete breakdown of the complexity of the LENS processing is presented in [3].

5.2 Results: Anechoic Environment

The anechoic environment simulations provide a good way to begin the discussion of hybrid processing performance: the predictable source radiation patterns in an anechoic room allow a relatively easy understanding of the system characteristics. The ULAs are the first set of systems to be tested in this environment and the non-ULA arrays are presented next.

5.2.1 ULAs

The following plots show the performance of both the original ULA based system and hybrids of varying configurations. Figure 5-1 shows the average gain versus number of elements for each TJR input level of -20, -10, and 0dB. Note that, because of the

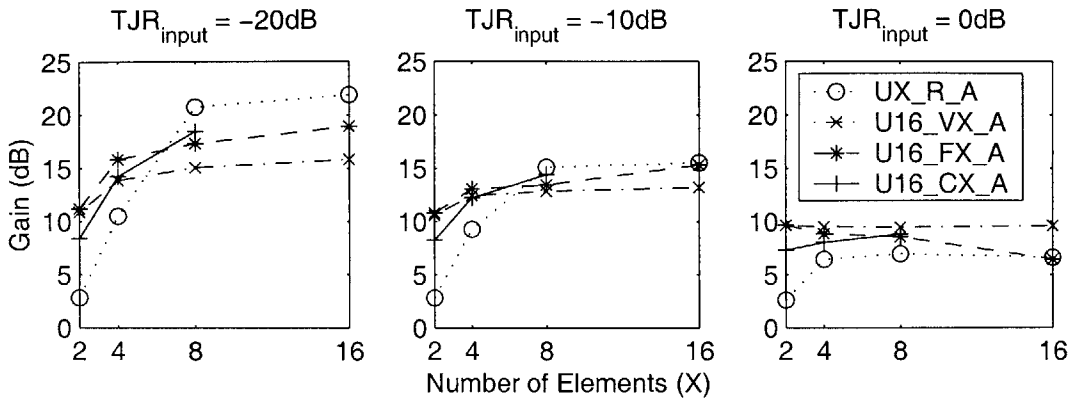

Figure 5-1: The average gain for each ULA based hybrid and the fully adaptive system versus the number of virtual elements in an anechoic environment.

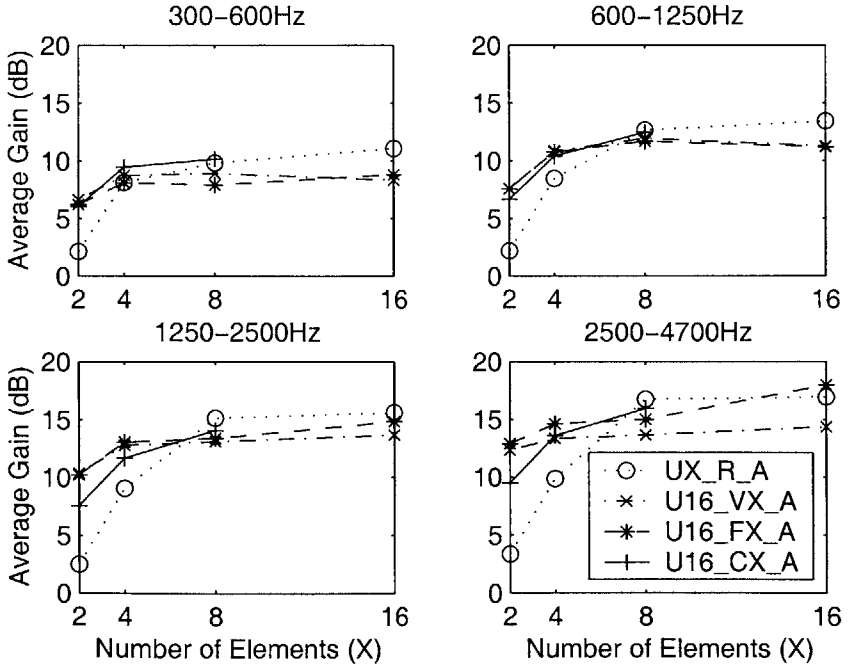

Figure 5-2: Plots of gain versus the number of elements for four frequency bands for the ULA in an anechoic environment with a TJR_{input} of -10dB.

architecture of the compressed array, there is no 16-virtual element hybrid under that configuration.

For all TJR input levels, the fully adaptive system brings the output TJR level to better than 0dB. However, the original system is not very robust as the TJR input level begins to increase. By the time the input TJR is at 0dB, there is little benefit to using a fully adaptive system. On the other hand, the hybrid systems maintain a more constant level of performance across all values of M'. For low values of TJR_{input} and $M' = 16$, the hybrid systems underperform the adaptive system, but as M' decreases, the performance of the hybrid systems falls off more slowly. For $M' = 4$, the hybrid systems all out perform the adaptive system. The same is true of increasing TJR_{input}, the hybrid systems prove to be more robust for all values of M' than the original system.

Figure 5-2 shows the performance for the different frequency bands with a TJR_{input} = -10dB. In almost all cases, the $M' = 16$-element virtual arrays underperformed the $M = 16$-element real ULA. Only at high frequencies could the fixed spacing hybrid out perform the real ULA system. Similar is true for the 8-element configurations. There is little hybrid performance drop as M' goes from 8 to 4 elements, as compared to the larger performance drop of the original system. This pushes all hybrid system performance levels above the fully adaptive system. Two-element hybrid performance is also better than the 2-element fully adaptive system ULA.

5.2.2 Non-ULA arrays

Presented in Figure 5-3 are the results of simulations of the circular and randomized arrays in an anechoic environment with TJR_{input} at -20dB, -10dB and 0dB. Since these arrays are likely to be seen in field applications, their results are of particular importance. There is no real array that can be directly processed by LENS for comparison with the hybrid systems, as with the ULAs. So as a baseline, the optimal fixed array will be used to compare the performance of the hybrid systems and this is shown as $M' = 1$ on all the plots.

The performance of each of the tested systems is very similar. It is odd that as

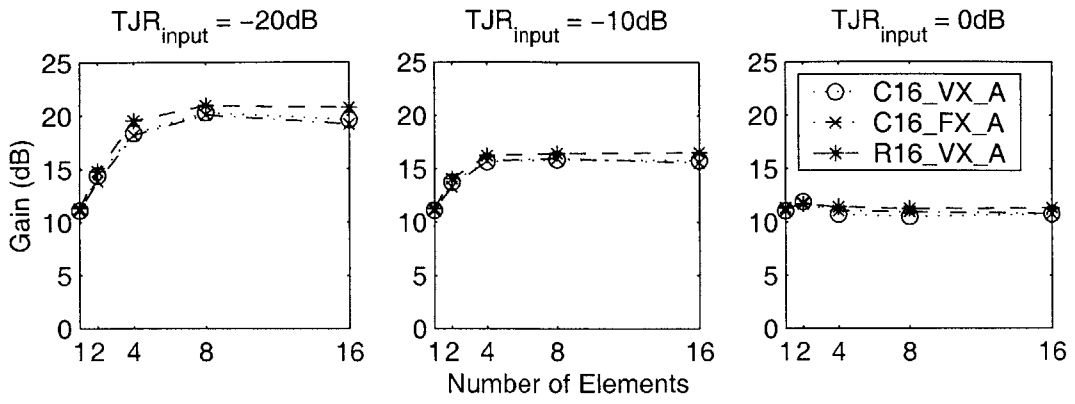

Figure 5-3: The average gain for the non-ULA hybrid systems versus the number of virtual elements in an anechoic environment.

the number of virtual elements increased to 16, each system seems to under perform its 8-element counterpart. The 4, 8, and 16-element hybrids perform about same as each other, and only with lower TJR inputs does the 2-element hybrid performance fall out of comparison. The performance of the fixed system is weak with low input TJRs, up to about 10dB lower performance than the 4-element hybrid system at TJR_{input}=-20dB. The baseline fixed system, is however, more robust in the presence of a strong target signal and the performance disparity disappears.

Broken down by frequency, in all except the high frequencies, the 2-element hybrid performs comparatively with the high order hybrid systems. This indicates that the difference between the $M' = 2$-element and $M' = 4$, 8, and 16-element systems shown in Figure 5-3 seems to be mostly evident between 2500-4700Hz.

These results suggest that there is much to be gained by going with a hybrid system, even with just 2 or 4 virtual elements. The performance of a 2-element hybrid over the fixed filter system is worthy of consideration given how little additional complexity would be added to the over all system.

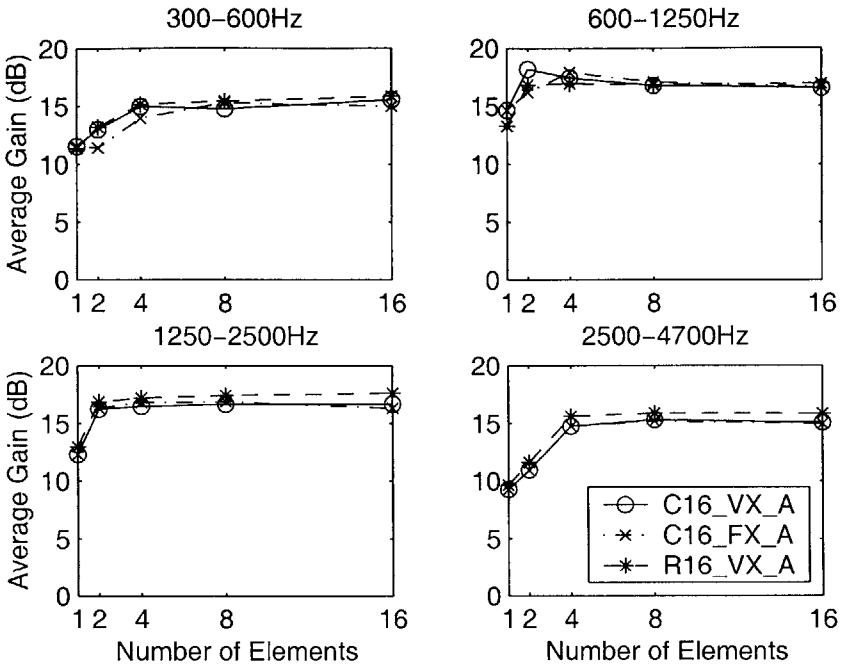

Figure 5-4: Plots of gain versus the number of elements for four frequency bands for the non-ULA arrays in an anechoic environment with a TJR_{input} of -10dB.

5.3 Results: Reverberant Environment

The tests in the reverberant environment are meant to give a better understanding of how the hybrid systems might perform in a real room environment. The tests were performed similarly to the anechoic cases. The results taken show strong hybrid performance, with them being more robust than the fully adaptive system for some scenarios.

5.3.1 ULAs

The performance of the ULA hybrid in a reverberant environment show promising results and are in-line with the performance of these systems in the anechoic environment. Figure 5-5 shows the break down of the hybrid performance versus TJR input. As in the anechoic case, the hybrid systems are more robust than the original system when $TJR_{input} = 0$dB, and now for lower TJR input, the hybrid systems with 8-virtual elements maintain better performance than the original system.

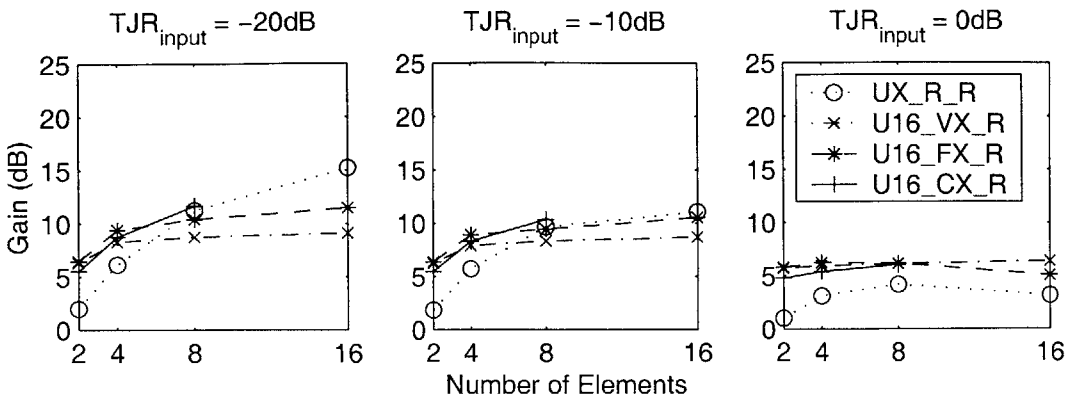

Figure 5-5: The average gain for each ULA hybrid system versus the number of virtual elements in a reverberant environment.

Figure 5-6 shows the performance of the hybrid ULA systems across frequency with a TJR input of -10dB. The results are good and show that the 8-element fixed spacing hybrid system is equivalent to the 8-element original system. For high frequencies, the fixed spacing hybrid system performs better than the original system.

5.3.2 Non-ULA Arrays

The fixed and variable spaced circular array and the variable spaced randomized array were tested at a TJR_{input} of -20dB, -10dB, and 0dB. Once again, the systems are compared to the optimal fixed filter system. Reverberant performance is quite consistent with expectation: similar performance for $M' = 4$, 8, and 16 virtual elements and slightly lower performance for $M' = 2$. Average TJR improvement for all hybrid systems is about 12dB, except when the TJR input was 0dB. At that level, the 2-element performs the best with the fixed system just behind. The additional adaptive freedom in higher order systems seems to make them less robust.

The performance is not quite as good as in the anechoic case, but still very respectable given similar performance differences between the ULA in the anechoic and reverberant environments. Figure 5-7 shows the performance plots of the non-ULA arrays in the reverberant environment.

Figure 5-8 shows the results for these array hybrids broken down by frequency at

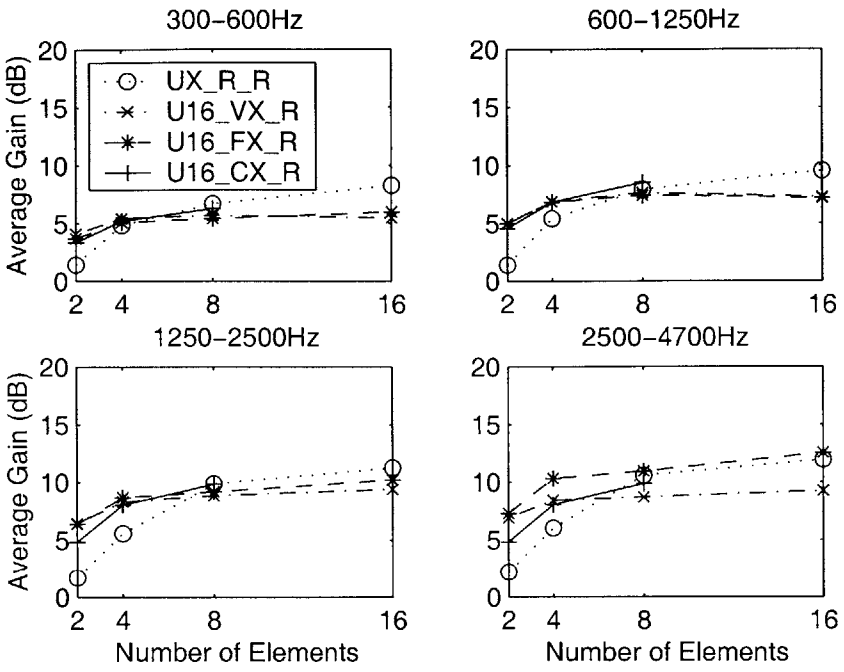

Figure 5-6: Plots of gain versus the number of elements for four frequency bands for the ULA in a reverberant environment with a TJR_{input} = -10dB.

a TJR_{input} of -10dB. The results are consistent across frequency, and are especially consistent across hybrid size of 4, 8, and 16. The 2-element hybrid is placed equally between the fixed system and higher order adaptive systems.

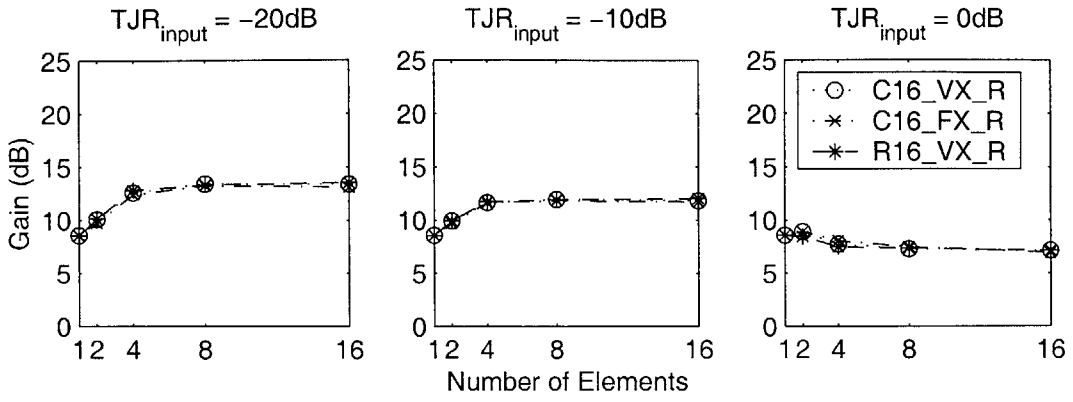

Figure 5-7: The average gain for each generalized hybrid system versus the number of virtual elements in a reverberant environment.

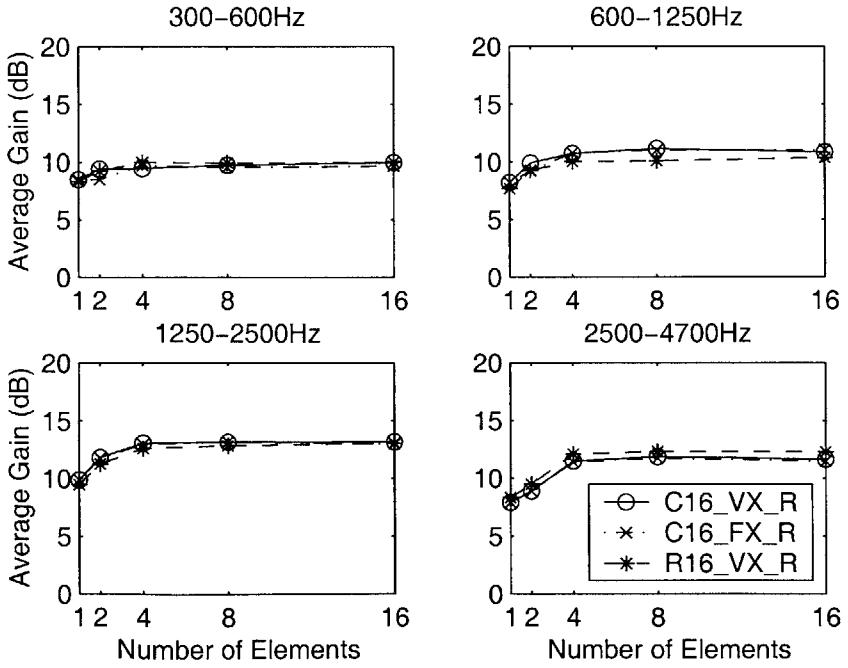

Figure 5-8: Plots of gain versus the number of elements for four frequency bands for the non-ULA arrays in a reverberant environment.

Chapter 6

Discussion and Conclusion

The results presented in the previous chapter suggest that there can be much benefit to using a hybrid system. Simulation results show strong performance with much less computation and increased robustness. This gives reason to further investigate and quantify the benefits of hybrid beamforming systems.

6.1 Hybrid Performance Observations

There are a number of interesting features of the hybrid system performance:

6.1.1 Good Performance with low *M'*

Hybrid systems provide comparable target enhancement to fully adaptive systems – even when the number of virtual elements M' is significantly lower than the number virtual elements. Specifically, with the circular and randomized array based systems, there is no apparent benefit to increasing M' beyond 4, as the performance versus number of element curves are flat after 2. Even just 2 virtual elements provided about 5dB better TJR improvement over the fixed filter system. Within certain frequency bands, moreover, near maximal performance is obtained with $M' = 2$. This maintenance of performance as M' decreases reflects the directionality of the virtual elements: since the virtual elements effectively use fixed processing to attenuate jam-

mer source, fewer adaptive degrees of freedom are required to attenuate the remaining jammer power.

6.1.2 Reduced Complexity

The primary reason to lower M' is to reduce the complexity of the overall system. As shown in Figure 2-5, the number of adaptive degrees of freedom greatly influences the amount of computation required. The hybrid systems tested show that the complexity can be reduced by orders of magnitude from the fully adaptive system. The strong performance of the hybrid systems for small M' show that systems with good performance can be achieved while greatly lowering complexity.

6.1.3 Increased Robustness

Another important benefit of lowering M' is that robustness is increased. As an example, Figure 5-1 shows that for high TJR_{input}, the hybrid systems are more robust to target cancellation. This feature is evident for nearly all hybrid configurations. Limiting the number of degrees of freedom of the adaptive system restricts its ability to cancel the target. Furthermore, in more reverberant environments, the directivity of the virtual elements effectively increases the direct-to-reverberant ratio, easing the task of jammer cancellation for the adaptive processor.

6.2 Future Work

The performance observations motivate several possible areas of future work.

6.2.1 Make the Virtual Elements More Similar

Overall hybrid performance can be increased by making the virtual elements more similar to each other. In the ULA based simulations, the compressed array performed better than the other hybrid systems in all but a small number of simulations. This performance difference can be attributed to the uniformity of the compressed array

elements. In the compressed virtual array, each of the elements had the same gain pattern, and the only interelement difference was the phase. In this case, even the phase maintains a pattern similar to that of a real array, inside and outside of the sector. In the other configurations, the phase and magnitude of the virtual elements was only constrained inside of the sector. Outside of the sector, the behavior was left unrestricted. This added difference complicated the input to the adaptive subsystem, thus decreasing its performance.

Future hybrids could include additional phase and magnitude constraints outside of the sector. This would reduce the complexity of the virtual array output and ensure a more predictable performance from the adaptive subsystem. The constraints would not necessarily need to make the virtual elements behave more like real elements, but make them more consistent.

6.2.2 Develop an Optimized Hybrid System

Looking at the more detailed performance plots in Appendix C shows that there are certain frequency bands where certain hybrid configurations work particularly well. It would make sense to be able to choose the hybrid system configuration based on the performance at each frequency.

For example, Figure 6-1 is the gain versus frequency plot for the fixed spacing circular hybrid. For the frequency band between 700 and 2200Hz, the 2-element hybrid performs about as well as the other hybrids. So for these frequencies, the 2-element hybrid could be used. For the other bands where the 2-element hybrid performance drops off, the 4-element hybrid could be used since its performance is similar to the 8-element hybrid. The net effect is a system that performs similarly to the 16 element hybrid, with a complexity that is less than the 4-element hybrid.

This is just one possible breakdown, and an infinite number of systems can be created based on different cost criteria. For example, some systems may need to be optimized for gain in the telephone voice band of 300-3000Hz, and thus could use a 4-element or better hybrid in that region, and use a fixed filter outside of that band.

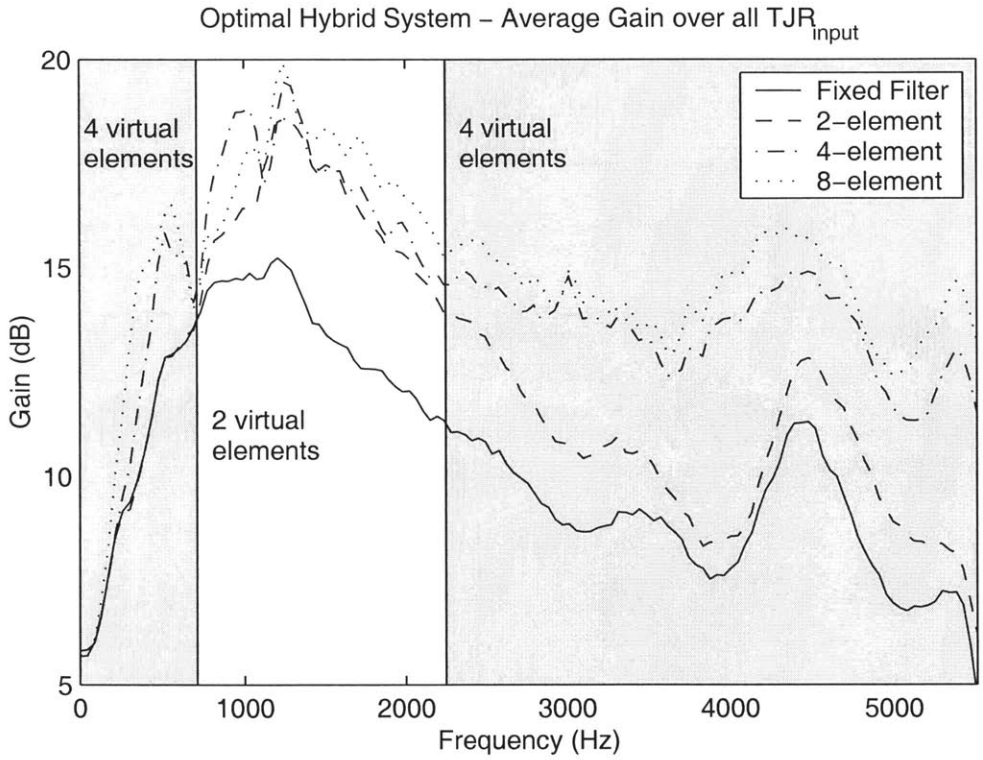

Figure 6-1: One possible optimal segmentation of a hybrid system based on the fixed circular hybrid

6.2.3 Additional Research Ideas

There are other ideas that might lead to further insight in the area. Work that might be interesting to pursue would be:

- Test hybrid systems using a virtual array with virtual elements spanning the real array. (Currently, only the 16-element virtual array has the same span as the real array.)
- Run simulations with a wider choice of jammers (i.e., using differing numbers, power levels, etc).
- Build a physical array and test some of these theories in anechoic and real room environments.

The motivation to pursue these additional ideas would be based on the desire to build an array processing system. The tools and the positive results presented in this thesis suggest that it is worthwhile to evaluate a hybrid system when designing an array processing system. The desired performance, the expected environment, and feasible array configuration will all effect the end performance and each of these issues needs to be considered on a per application basis. These additional ideas will help evaluate the suitability of such a system.

6.3 Conclusions

With the simulation results obtained in this thesis, there is strong motivation to continue work in this area. The type of hybrid system presented here can be a useful tool when designing an array processing system, allowing the designer to reduce the complexity of the overall system while maintaining an respectable level of performance and increasing robustness. These are factors not as easily controlled in either a fully fixed or fully adaptive system alone, but are now in the hands of the designer.

Appendix A

Virtual Array Figures

Following is a set of figures showing the gain and phase plots of the virtual arrays used for testing. The plots given are for select configurations only since each hybrid configuration has 129 sets of virtual elements, one for each frequency bin.

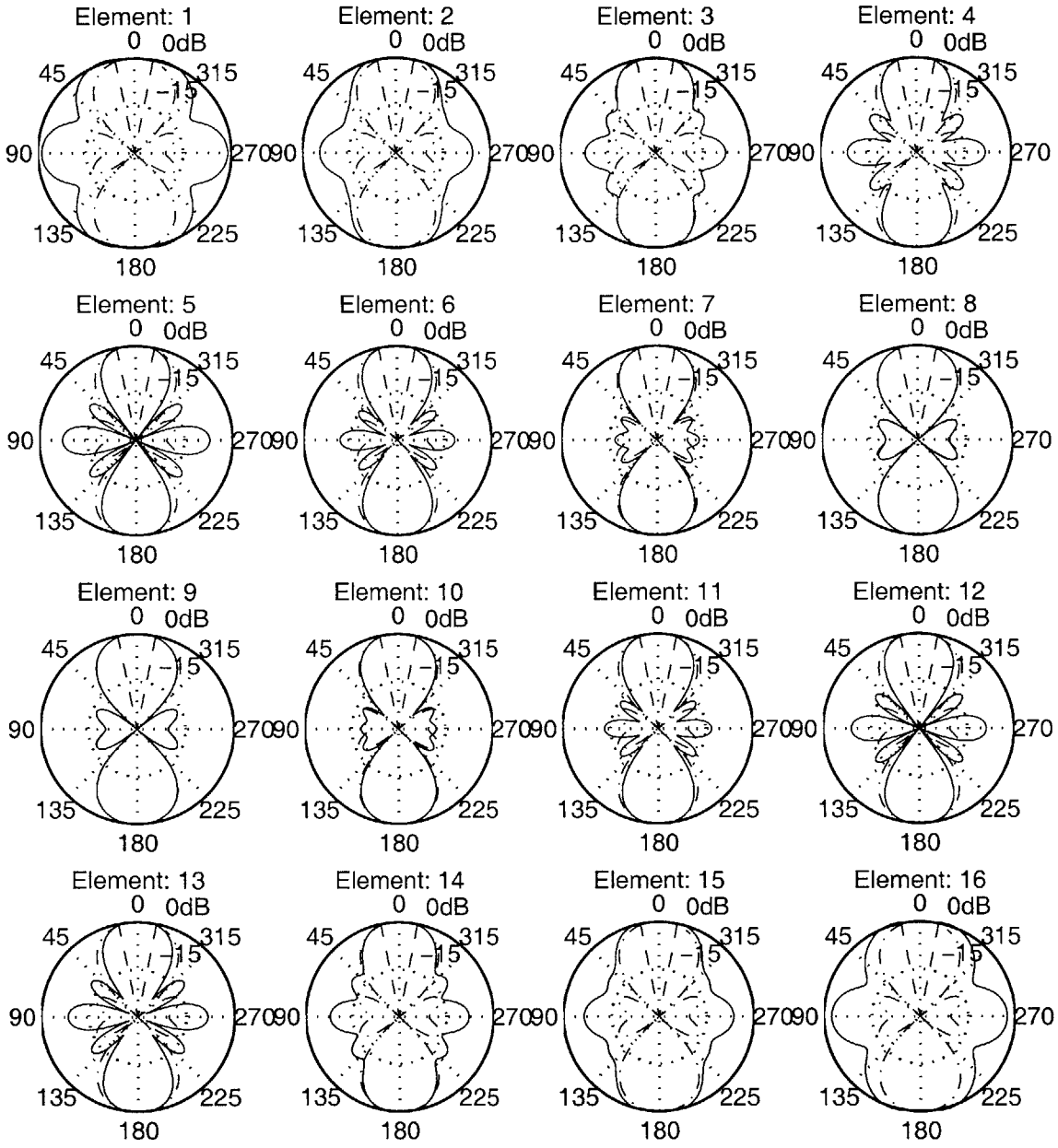

Figure A-1: Gain patterns for the 16 element virtual array based on the ULA at 1000Hz with variable virtual element spacing.

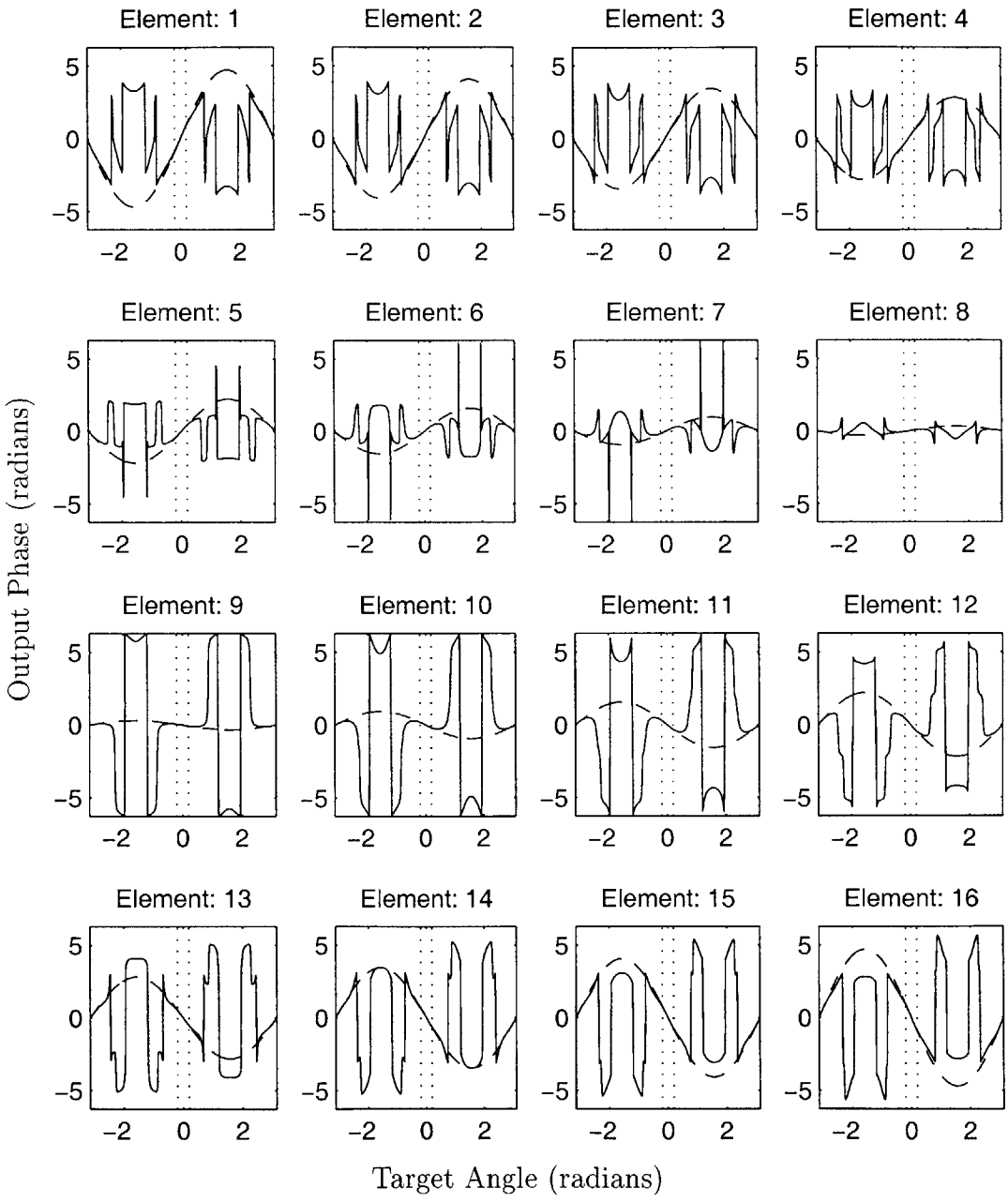

Figure A-2: Phase plots for the 16 element virtual array based on the ULA at 1000Hz with variable virtual element spacing.

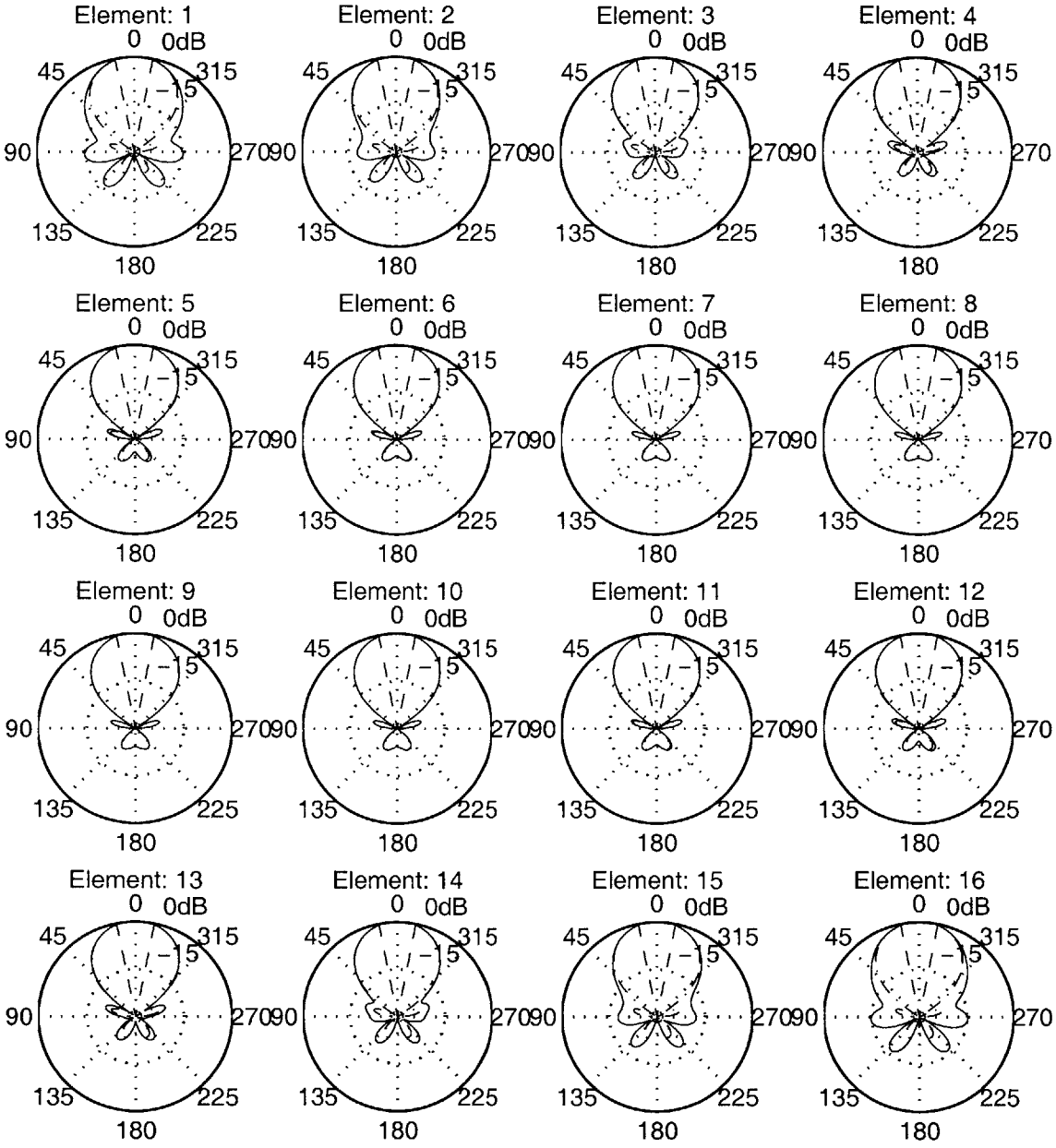

Figure A-3: Gain patterns for the 16 element virtual array based on the circular array at 1000Hz with fixed virtual element spacing.

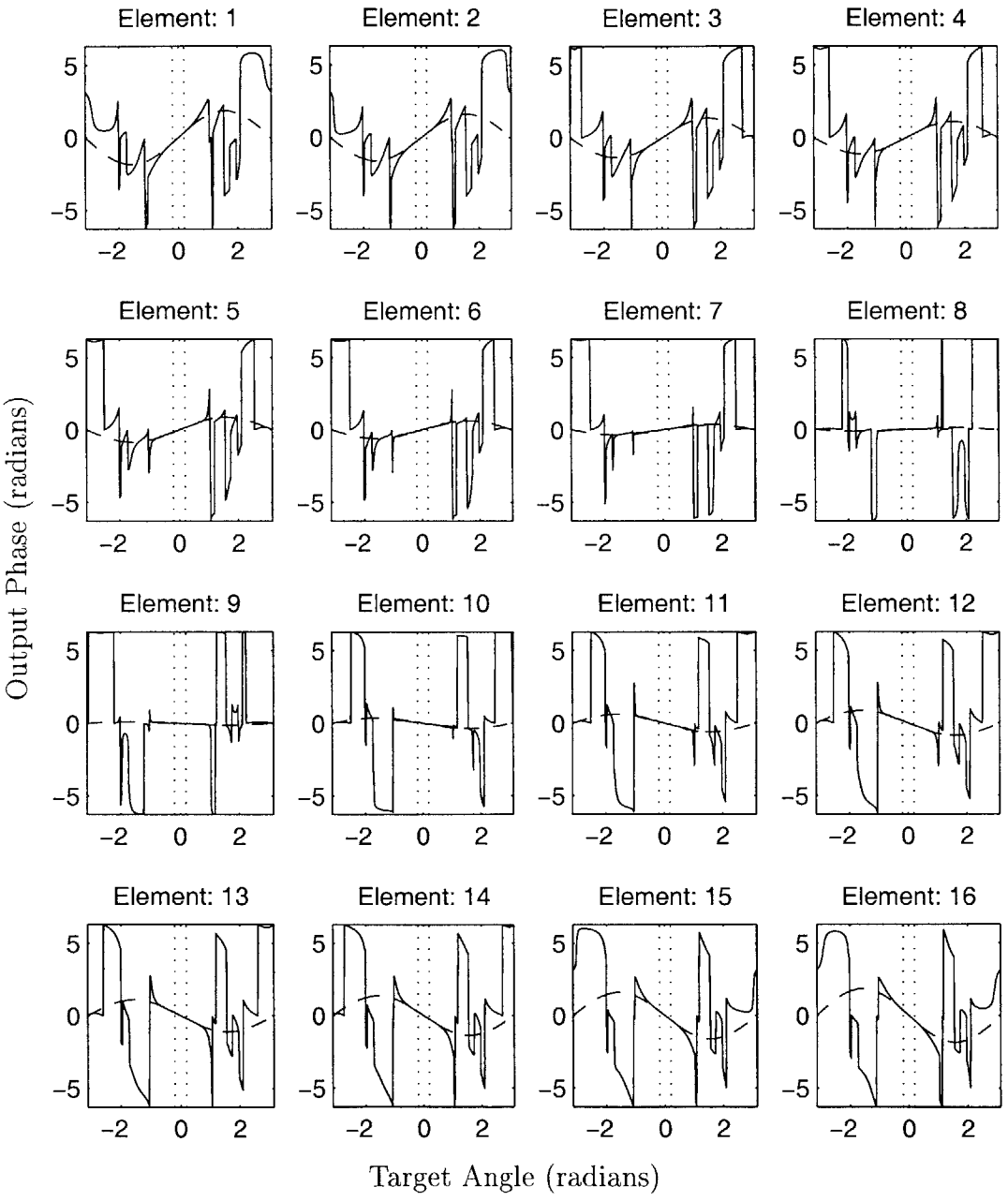

Figure A-4: Phase plots for the 16 element virtual array based on the circular array at 1000Hz with fixed virtual element spacing.

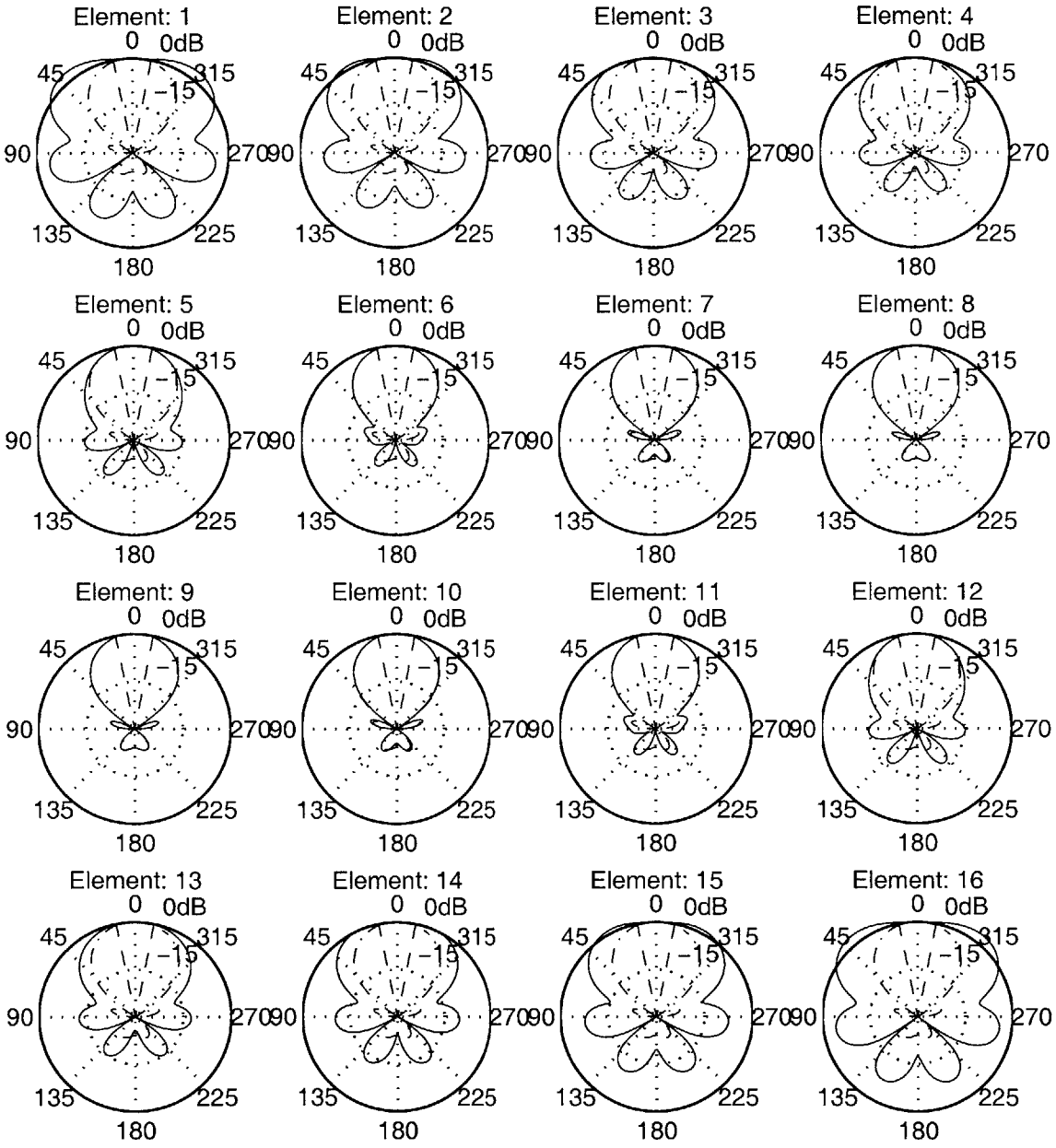

Figure A-5: Gain patterns for the 16 element virtual array based on the circular array at 1000Hz with variable virtual element spacing.

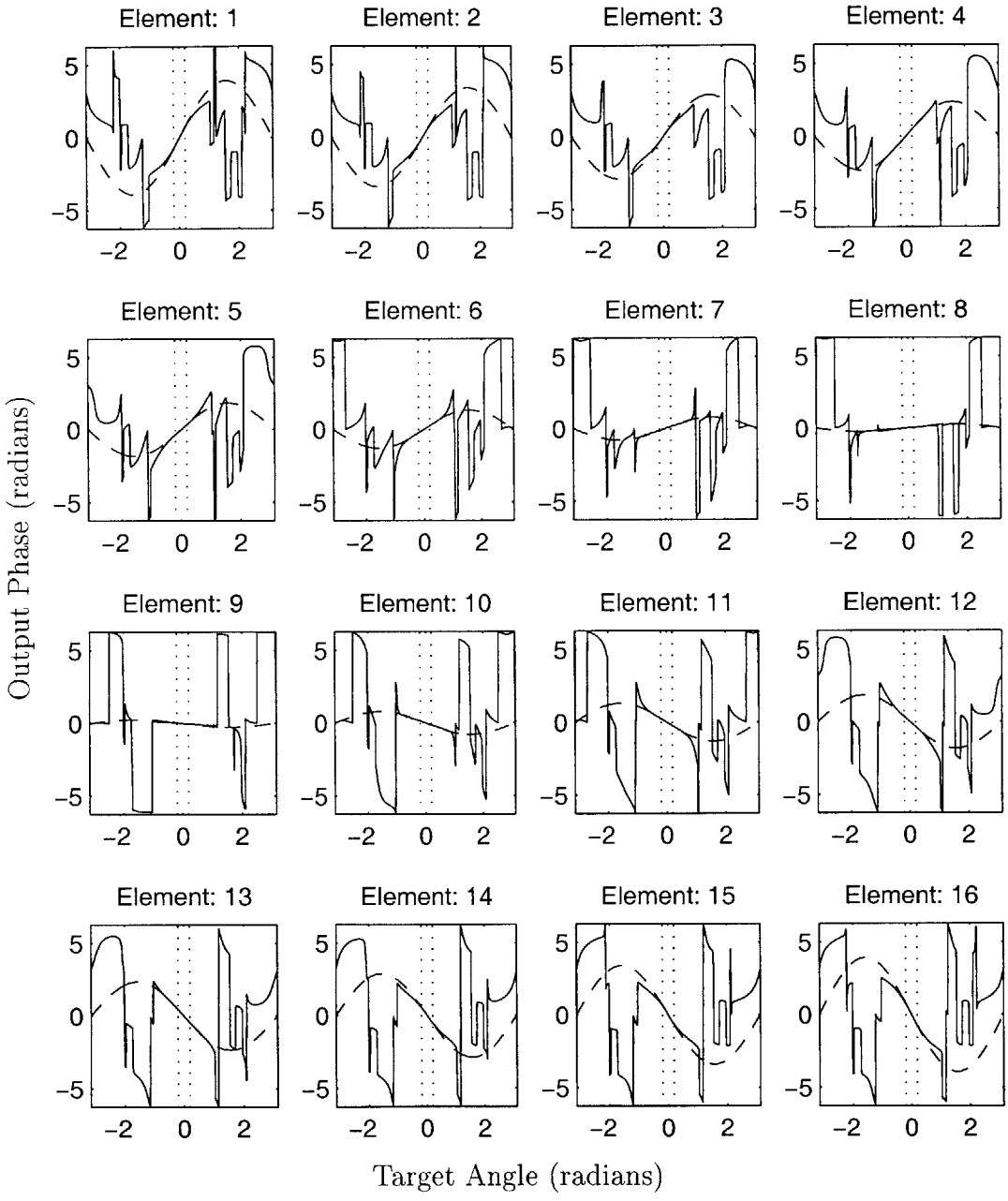

Figure A-6: Phase plots for the 16 element virtual array based on the circular array at 1000Hz with variable virtual element spacing.

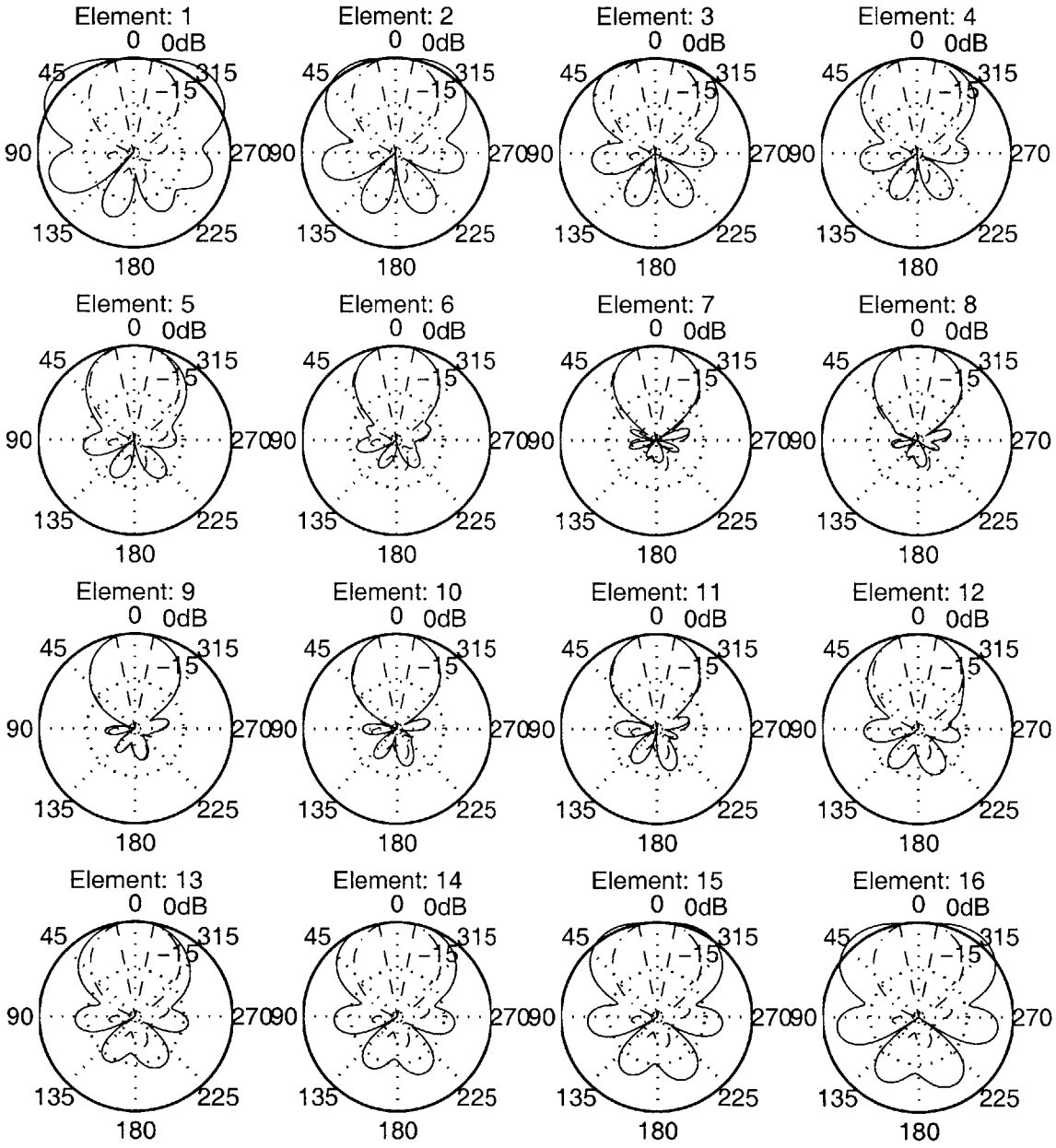

Figure A-7: Gain patterns for the 16 element virtual array based on the randomized array at 1000Hz with variable virtual element spacing.

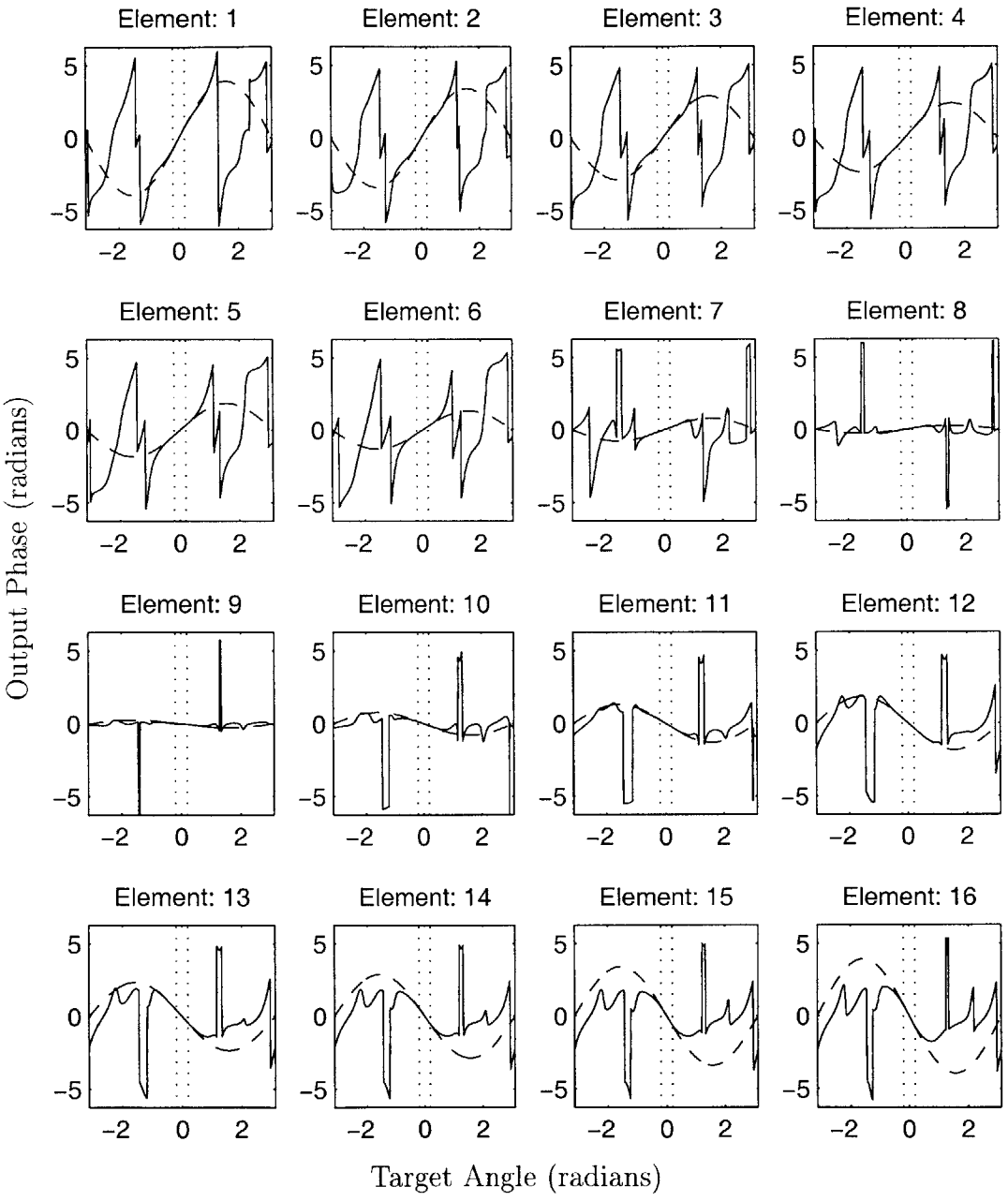

Figure A-8: Phase plots for the 16 element virtual array based on the randomized array at 1000Hz with variable virtual element spacing.

Appendix B

Simulation Details

This appendix lists the specific simulation parameters used in each of the tests. There is data for the environment and source configurations.

B.1 Environment Configuration

Each simulation was done with in either the anechoic or reverberant room. The anechoic room did not simulate any source reflections, only the direct source-to-array signal was included. For the reverberant room, the sources were allowed to reflect off the walls. The specific configuration is as follows:

- Room Dimensions: 6m x 5m x 2.5m
- Array Center location: [2.5m 2.8m 1.2m]
- Wall reflection coefficient: 0.4
- Impulse response length: 1000 taps
- Direct-to-Reverberant ratio: -.56dB

B.2 Sources

As stated earlier, 10 sets of source locations were chosen randomly. Each set was then used individually when testing the performance of a system. Figure B-1 shows the location of each of the test sources. Each source was placed 2 meters from the center and the array. The target source was place in the center of the target sector. Notice the locations of each of the jammer sources. this represents a reasonable sampling, with some sources just outside of the sector, others directly opposite, and some are clustered together, while others are spread more evenly.

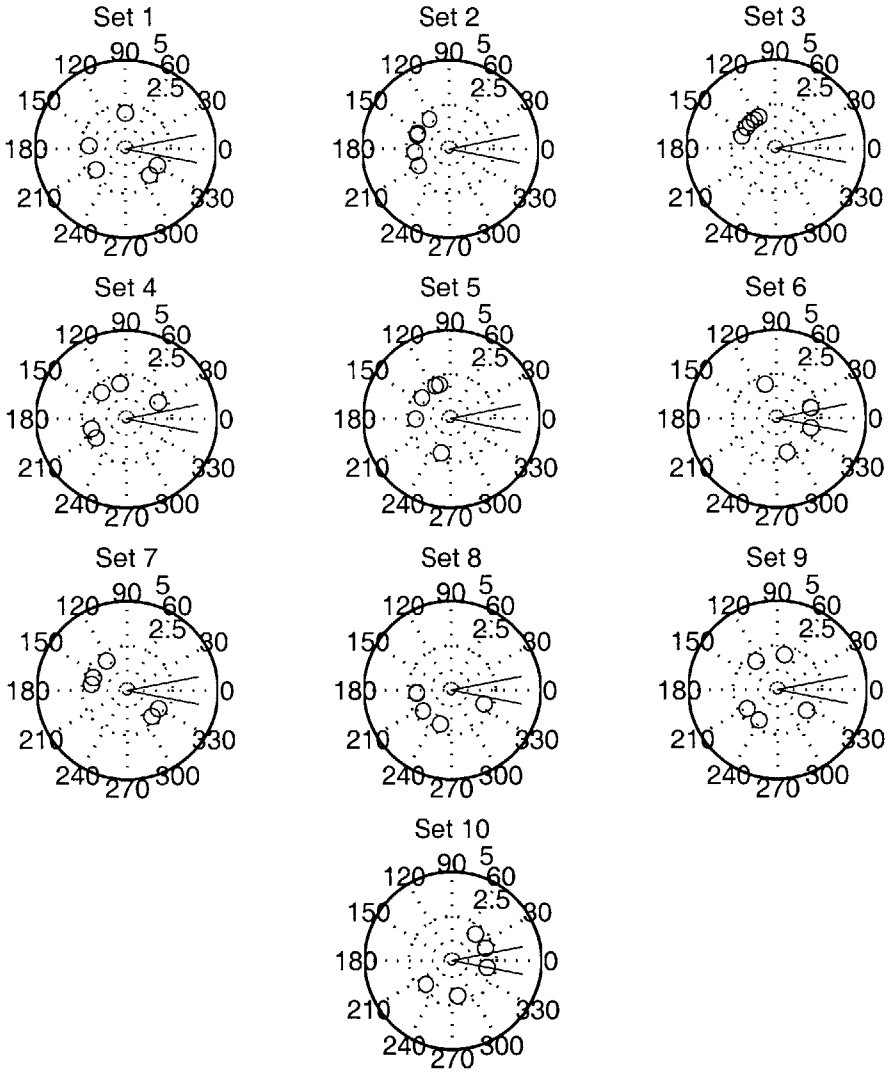

Figure B-1: Plot showing the source locations used for each of the ten iterations. Target Cell centered around 0 degrees.

Appendix C

Results

Presented in this appendix are the detailed plots of the each hybrid systems performance versus frequency. The ULA based systems are plotted against the fully adaptive system of the same complexity. This is because the adaptive system was used as a benchmark for comparison. Since the generalized systems have no such parallel benchmark, they will be plotted again the fixed filter performance as done in Chapter 5.

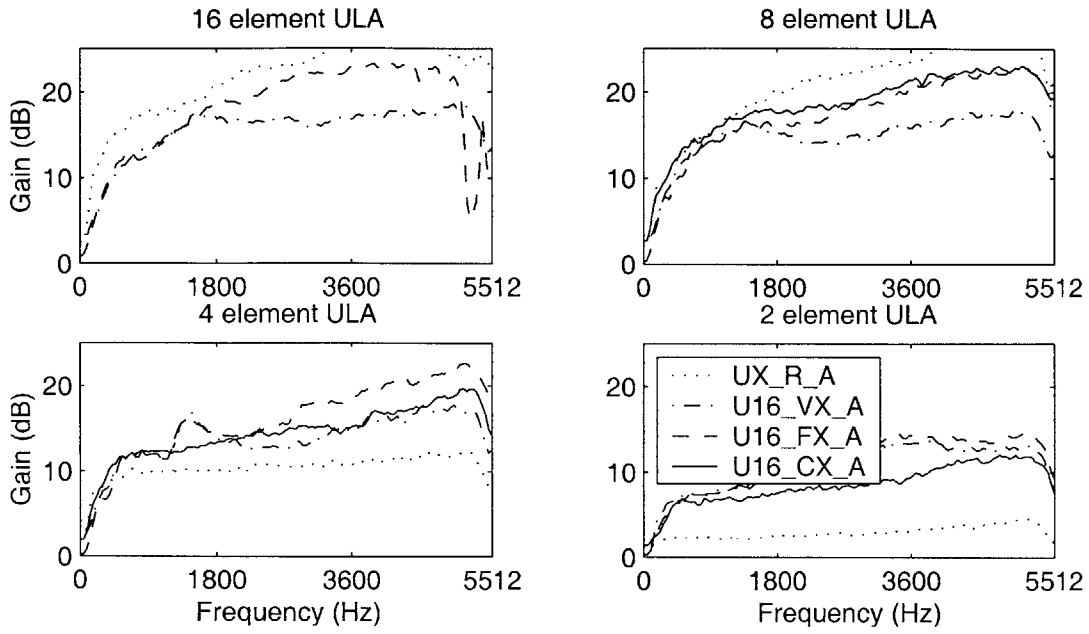

Figure C-1: ULA performance vs. frequency in an anechoic environment TJR input -20dB.

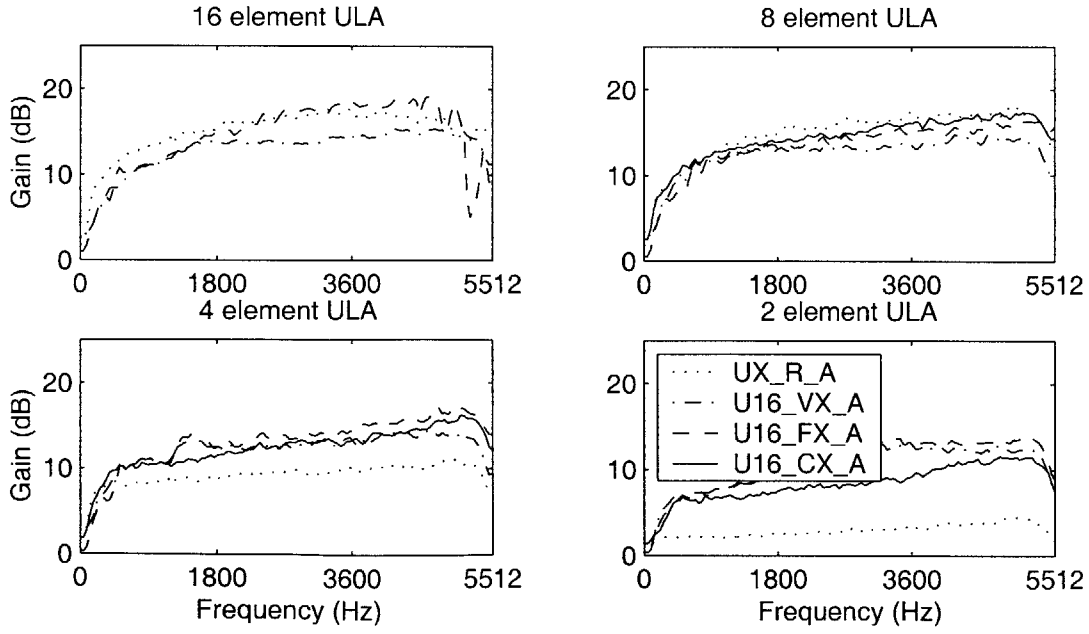

Figure C-2: ULA performance vs. frequency in an anechoic environment TJR input -10dB.

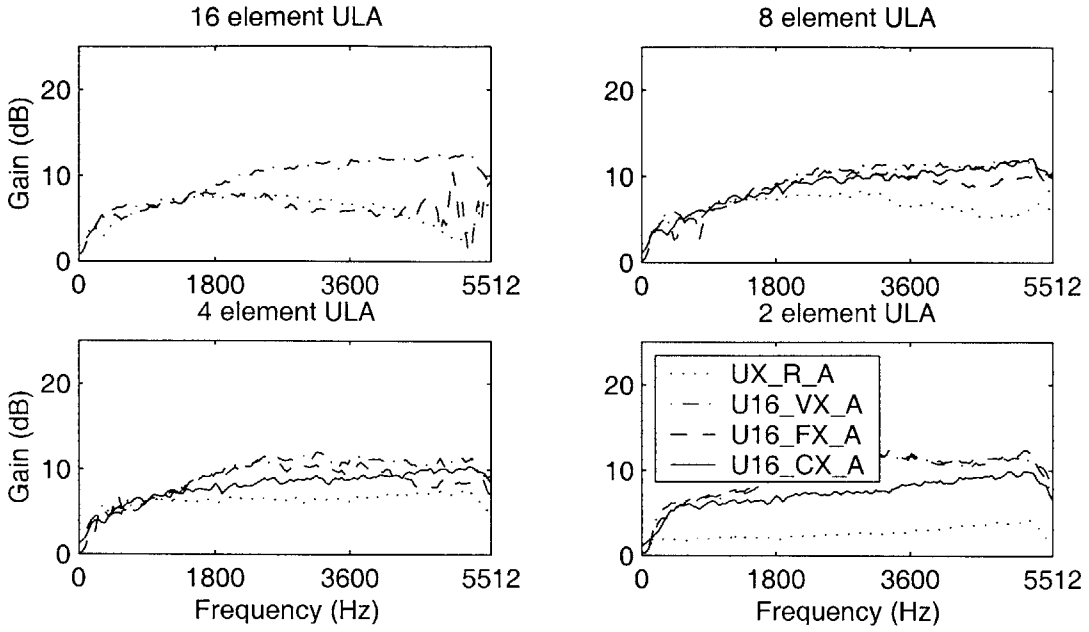

Figure C-3: ULA performance vs. frequency in an anechoic environment TJR input 0dB.

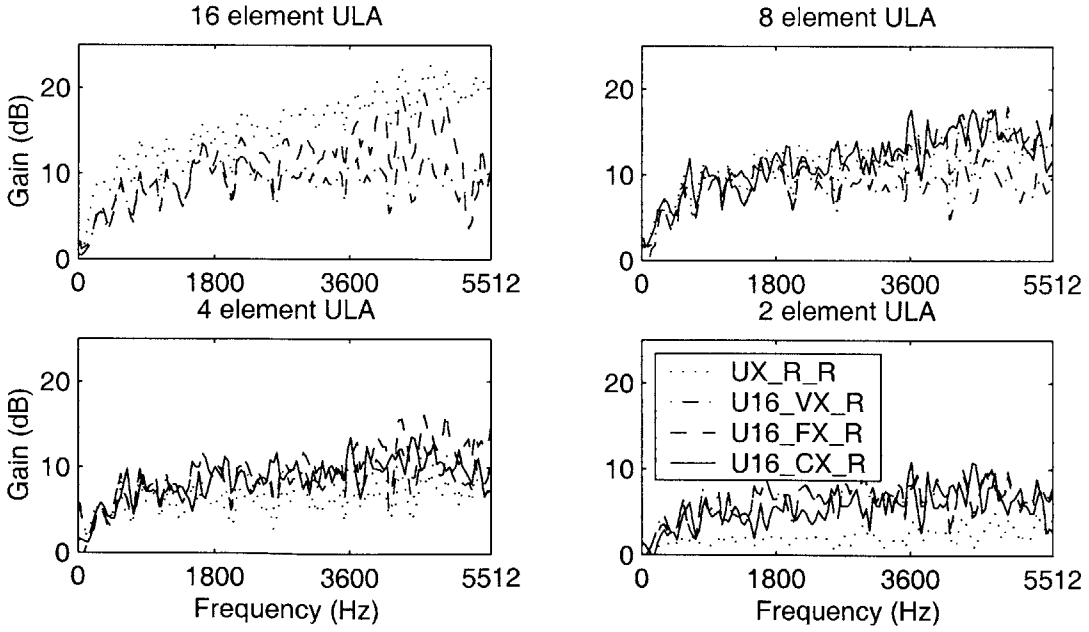

Figure C-4: ULA performance vs. frequency in an reverberant environment TJR input -20dB.

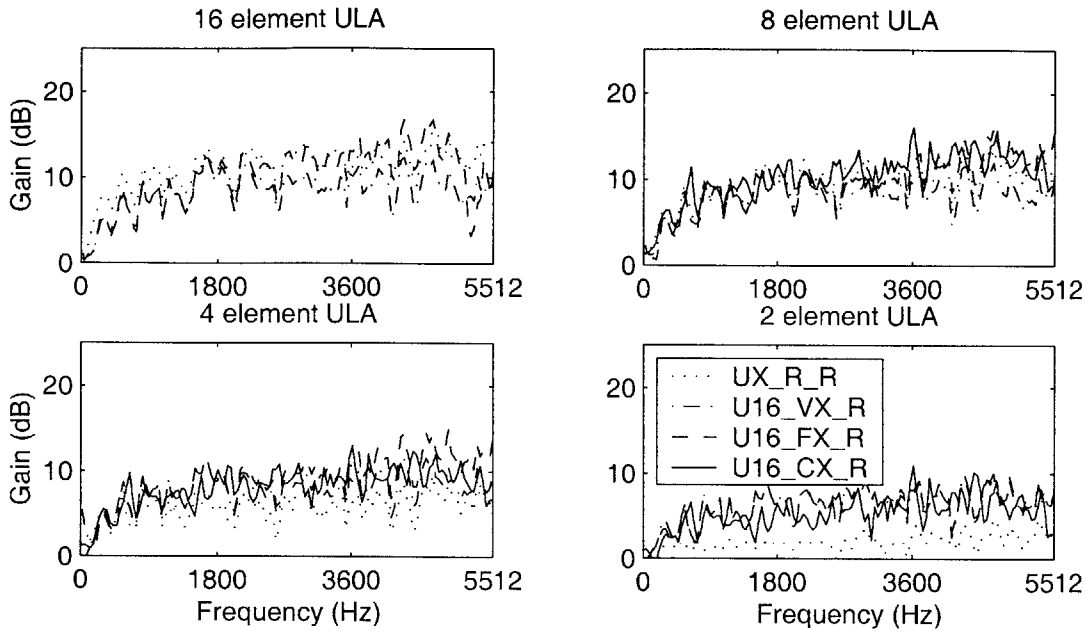

Figure C-5: ULA performance vs. frequency in an reverberant environment TJR input -10dB.

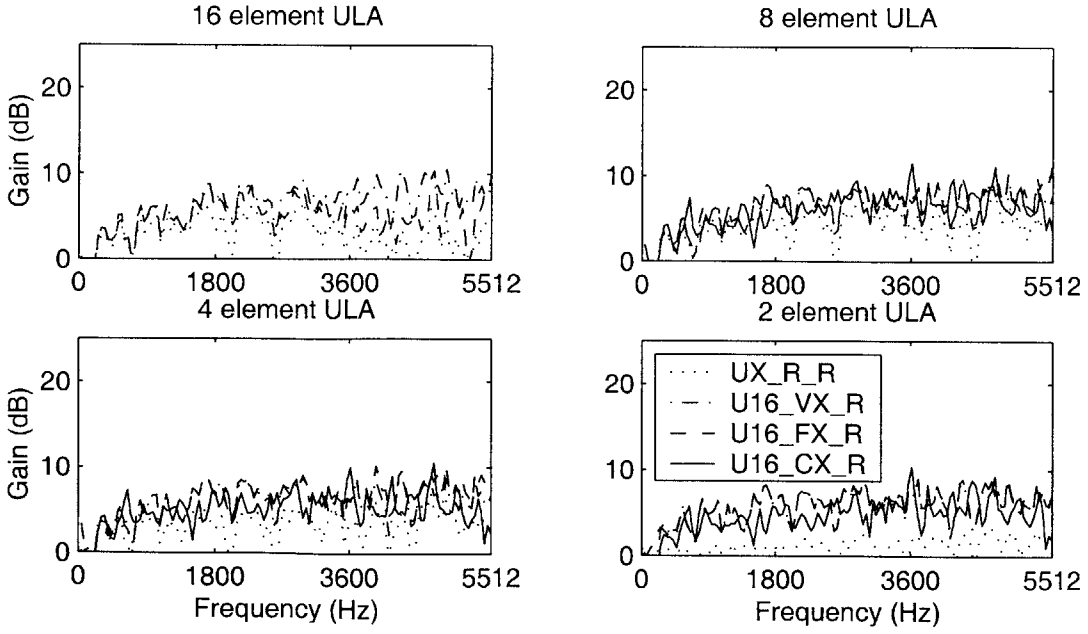

Figure C-6: ULA performance vs. frequency in an reverberant environment TJR input 0dB.

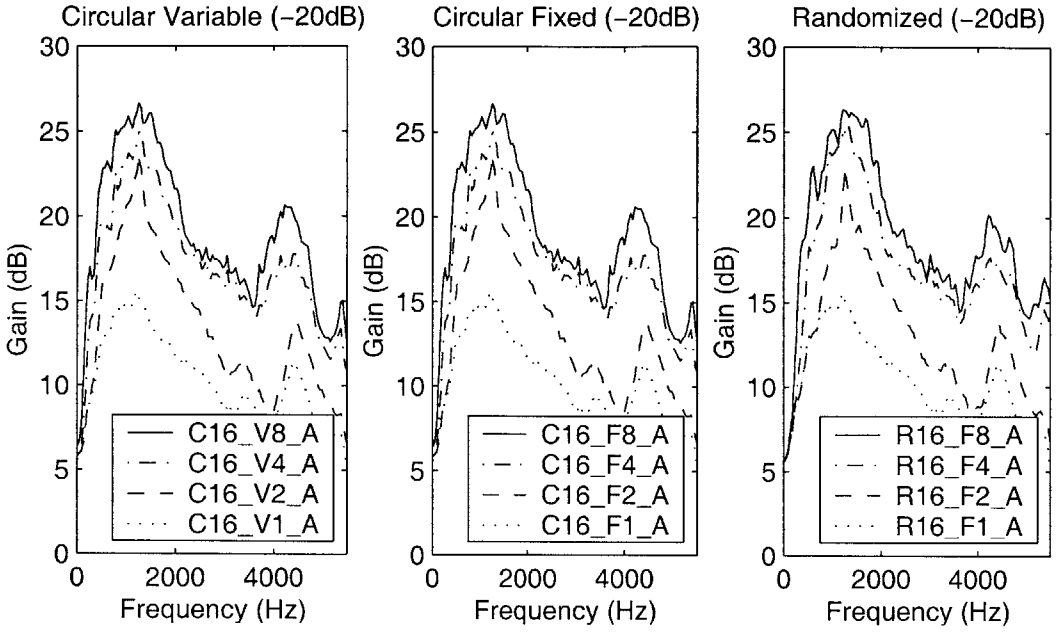

Figure C-7: Generalized array performance vs. frequency in an anechoic environment TJR input -20dB.

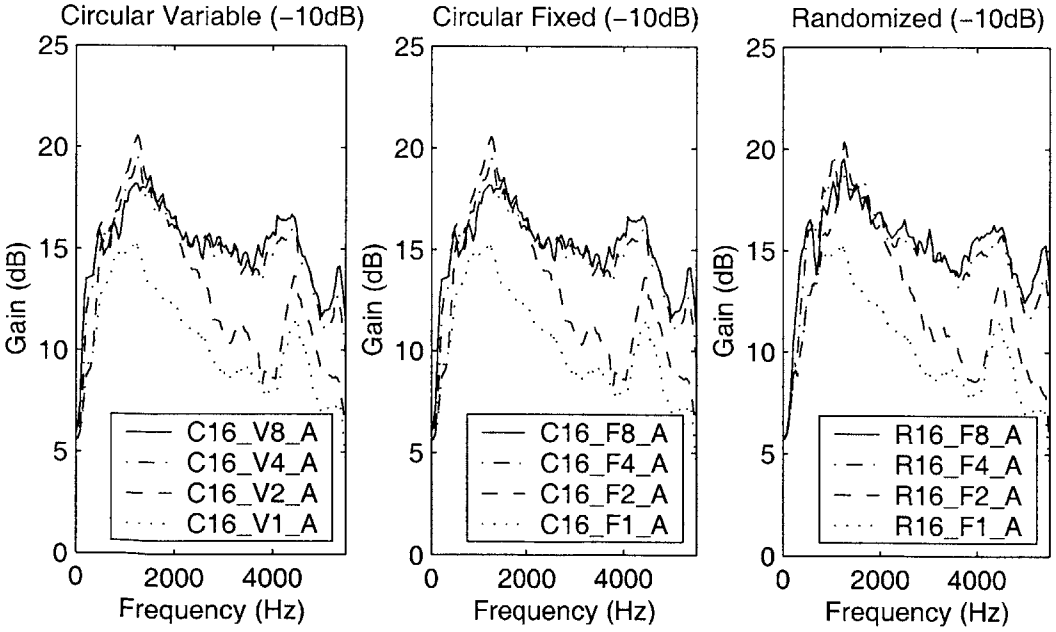

Figure C-8: Generalized array performance vs. frequency in an anechoic environment TJR input -10dB.

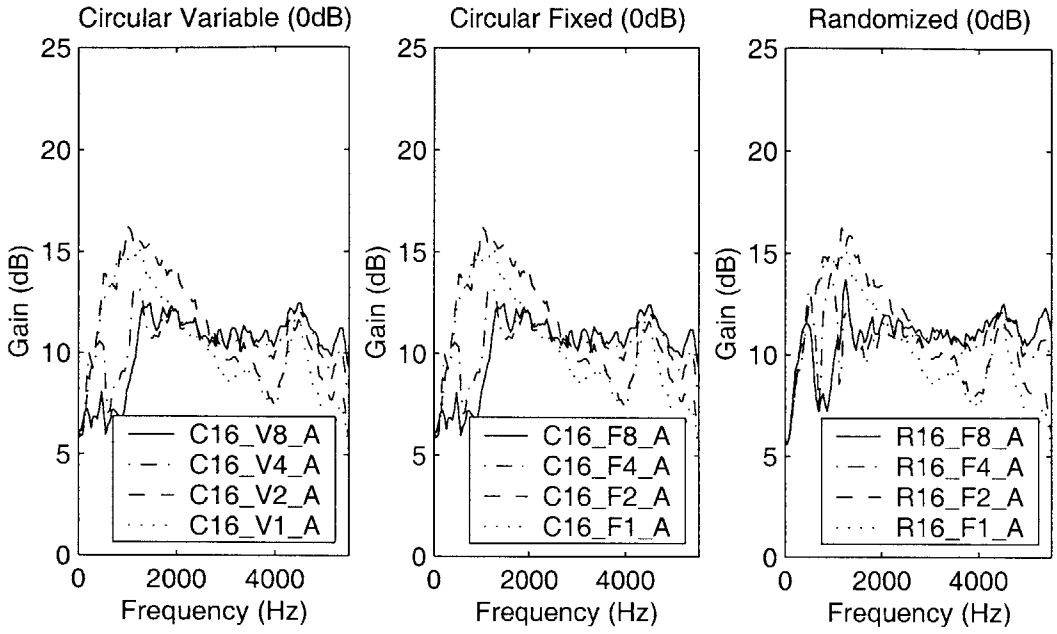

Figure C-9: Generalized performance vs. frequency in an anechoic environment TJR input 0dB.

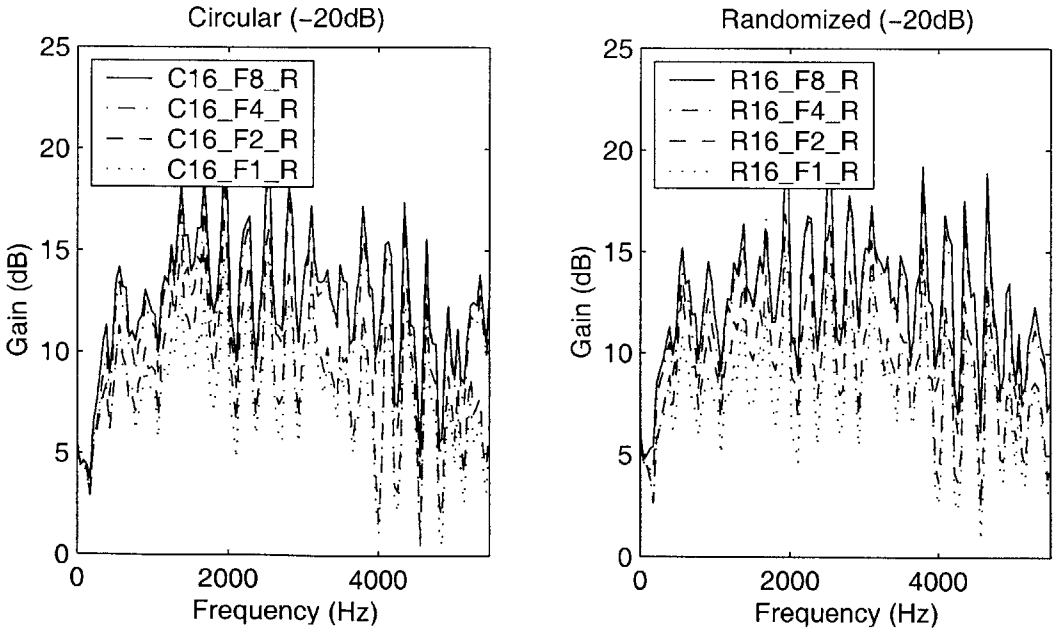

Figure C-10: Generalized array performance vs. frequency in an reverberant environment TJR input -20dB.

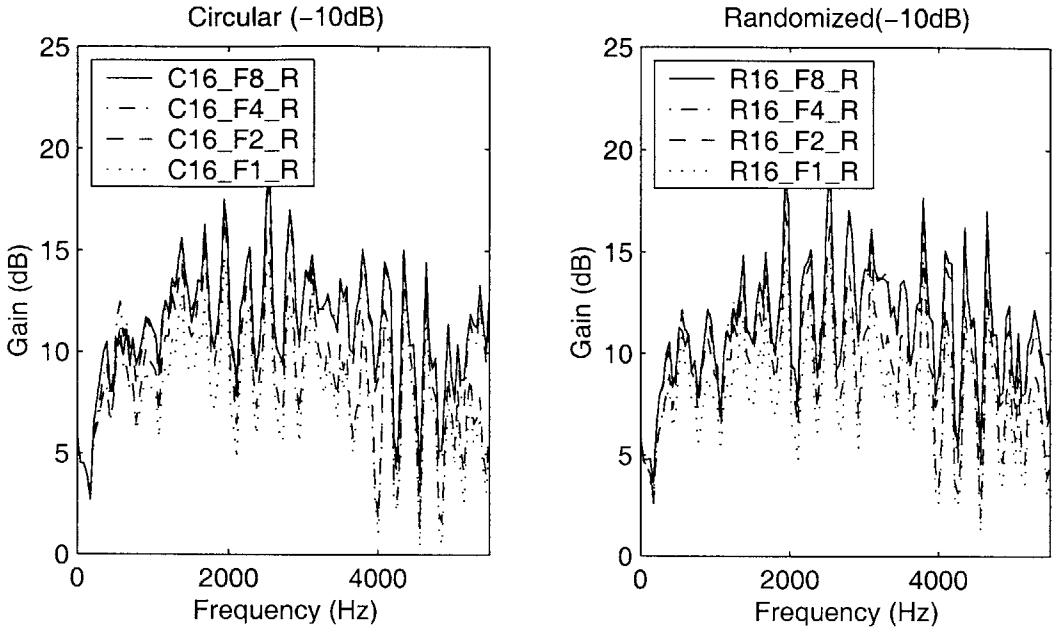

Figure C-11: Generalized array performance vs. frequency in an reverberant environment TJR input -10dB.

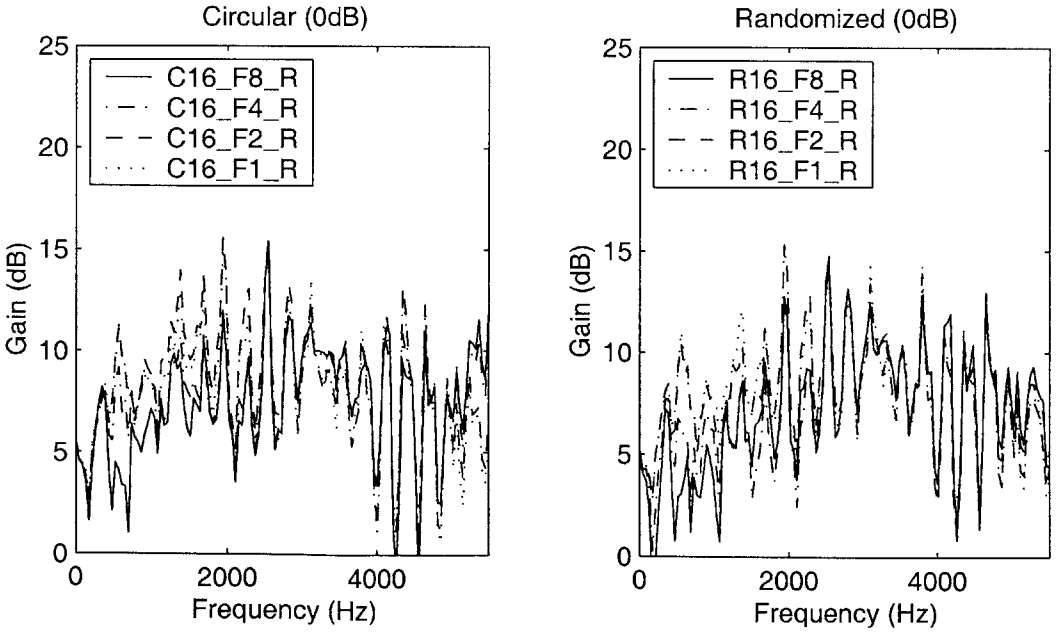

Figure C-12: Generalized array performance vs. frequency in an reverberant environment TJR input 0dB.

Bibliography

[1] D. K. Cheng and F. I. Tseng. Gain optimization for arbitrary antenna arrays. *IEEE Trans. Antennas Propag.*, AP-13:973–974, 1965.

[2] J. G. Desloge D. P. Welker, J. E. Greenber and P. M. Zurek. Microphone-array hearing aids with binaural output–Part II: two-microphone adaptive system. *IEEE Transactions on Speech and Audio Processing*, 5(6):543–551, November 1997.

[3] Joseph G. Desloge. *Location-Estimating, Null-Streering (LENS) Algorithm for Adaptive Microphone-Array Processing.* PhD dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, September 1998.

[4] S. Gazor and Y. Grenier. Optimal positioning of sensors for a microphone array. *Acoustics, Speech, and Signal Processing*, 4:557–560, April 1995.

[5] R. M. Zeskind H. Cox and T. Kooij. Pratical supergain. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-34:393–398, 1986.

[6] W. M. Rabinowitz J. G. Desloge and P. M. Zurek. Microphone-array hearing aids with binaural output–Part I: fixed processing systems. *IEEE Transactions on Speech and Audio Processing*, 5(6):529–542, November 1997.

[7] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[8] G. Valintini M. Kompis and M. Pelizzone. A combined fixed/adaptive beamforming noise-reduction system for hearing aids. *Proc. 20th Annual IEEE Engineering in Medicine and Biology Soc.*, 20(6):3136–3139, 1998.

[9] W. M. Rabinowitz P. M. Peterson, S. Wei and P. M. Zurek. Robustness of an adaptive beamforming method for hearing aids. *Acta Otolaryngol Suppl.*, 469:85–90, 1990.

[10] R.W. Stadler and W. M. Rabinowitz. On the potential of fixed arrays for hearing aids. *Journal of the Acoustical Society of America*, 94(3):1332–1342, September 1993.

[11] B. D. Van Veen and K. M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, pages 4–24, April 1988.