Collaborative Transmit Beamforming

A Thesis

Submitted to the Faculty

of

Drexel University

by

Daniel Saul Zalkind

in partial fulfillment of the

requirements for the degree

of

Master of Science in Electrical and Computer Engineering

May 2014

Acknowledgements

I'd like to thank the Drexel community and the ECE department for giving me every opportunity to pursue my goals; especially my adviser, Moshe Kam, who piqued my interests and provided guidance, support and advice.

I appreciate everyone I have met in and around the pool for preserving my sanity and helping me maintain balance in multiple aspects of life.

This thesis would not have been possible without hard work and dedication, traits I am grateful to have learned from my parents, in addition to the love and support I've received from all my family and friends.

Table of Contents

List of Tables

List of Figures

Abstract

Collaborative Transmit Beamforming

Daniel Saul Zalkind

Advisor: Moshe Kam, Ph.D.

Collaborative transmit beamforming is a form of communication where multiple transmitters cooperate to send a common message. If each transmitter properly controls the phase of its signal, the signals interfere constructively at the destination, translating in increased range, rate and energy efficiency. Collaborative beamforming has the potential to improve security and interference reduction by attenuating the signal transmitted in unintended directions. There are several challenges in collaborative transmit beamforming, like frequency synchronization, information sharing between nodes, and beampattern performance. This review focuses on the performance of collaborating nodes in terms of beampattern statistics when the transmitters are realized as a distributed ad hoc sensor network, and the nodes are randomly distributed over an area (Ochiai 2005, Mudumbai 2007). We review the design parameters necessary for the network to act as a distributed beamformer and support the finding of (Ochiai 2005) by simulations. The simulations treat the transmitting nodes as a virtual antenna array and collect relevant beampattern statistics.

1. Introduction

This thesis reviews the design and performance of a system of distributed transmitters collaborating to send a common message. Transmitters distributed in space can achieve the same performance as an antenna array by exploiting the space-time characteristics of a radiated signal. The improvements include enhanced directional performance and interference reduction. We study the system of distributed transmitters within the framework of a virtual antenna array, where each transmitter acts as a separate entity generating its own signal. Each transmitter control its phase so the system transmits a coherent signal in a desired direction. We study the performance of the system of transmitters and model the transmission of signals from them.

We present this study in the context of wireless sensor networks (WSNs), with nodes randomly distributed in space. If each node is equipped with a transmitter, collaborative beamforming allows a group of N transmitters to achieve a N^2-fold gain in signal power. This leads to increased transmission range (N-fold increase for free space propagation), increased rate (an N^2-fold increase in a power-limited regime), or increased energy efficiency (an N-fold decrease in the net transmitted power for a fixed desired receive power) [13]. Collaborative beamforming also allows the network to reduce interference and improve security.

1.1 Overview of Challenges

There are a number of challenges involved in transmitting a signal from a system of distributed transmitters. Using the framework of a virtual antenna array [4], we are able to study the system of distributed transmitters and describe their performance in the same manner as conventional antenna arrays. However, a number of differences

must be accounted for.

In conventional antenna arrays, elements are fixed in space and take a specified geometry to provide the baseline performance of the array [27]. Our transmitters are randomly distributed in space, so we studied the beamforming performance of random arrays.

Conventional antenna arrays are connected to a signal source through a feed network, ensuring each element transmits a version of the same signal. In our system, each transmitter generates its own signal, so we reviewed methods designed to synchronize the transmitters' carrier phase and frequency. Additionally, the transmitted information must be shared amongst nodes prior to collaborative beamforming. A method [5] has been developed in which the transmitters share and schedule the information to be transmitted.

1.2 System Model

To study the performance of distributed transmitters, we have done the following:

1. Developed a simulation framework, which generates the beampattern of virtual antenna arrays in different configurations and computes antenna parameters that describe their performance.

2. Review previously researched synchronization methods and determine the amount of error our the system will experience due to node synchronization.

3. Simulated random arrays and compared results with known measures [17].

4. Designed and simulated a system in which collaborative beamforming could be utilized to achieve performance gains.

1.3 Organization

This thesis is organized as follows.

Chapter 2 provides background knowledge of antenna arrays, which can be extended to virtual antenna arrays, and RF transmission analysis.

Chapter 3 details the simulation framework used to study virtual antenna arrays and measure their performance.

Chapter 4 reviews research in the synchronization of distributed transmitters.

Chapter 5 reviews the performance of a system of distributed transmitters.

Chapter 6 details the design of a system of distributed transmitters for a specific environment. We discuss design tradeoffs and alternative design options.

Chapter 7 concludes this thesis.

2. Background

2.1 Introduction

Because our system of distributed transmitters behaves as a virtual antenna array, it is useful to review antenna array characteristics and how a signal transmitted from one behaves. An antenna array is a collection of antennas that work together to act as a single transmitter. Antenna arrays offer improved directional performance and the ability to electronically steer its beam by controlling the signal in each element.

In this chapter, we review how multiple elements combine to form a beam. Then, we introduce terminology used to describe antenna beampatterns and parameters used to describe their performance. We review the transmission of signals in space and conclude the chapter with an example illustrating the discussed topics.

2.2 Array Formulation

Signals are radiated through space when a current is oscillated in a conductor. The conductor generates an electromagnetic field with strength $S(\phi, \theta)$ in all azimuth and elevation angles, ϕ and θ, respectively. This function is referred to as the element's *radiation pattern*, and depends on the antenna's properties.

Now consider multiple elements, where the signal radiated from element i has the radiation pattern $S_i(\phi)$. For simplicity, we will only consider transmission in the azimuth direction. Combining the radiation pattern of all elements gives the antenna array radiation pattern

$$S(\phi) = \sum_{i=1}^{N} S_i(\phi) \tag{2.1}$$

where N is the number of elements in the array.

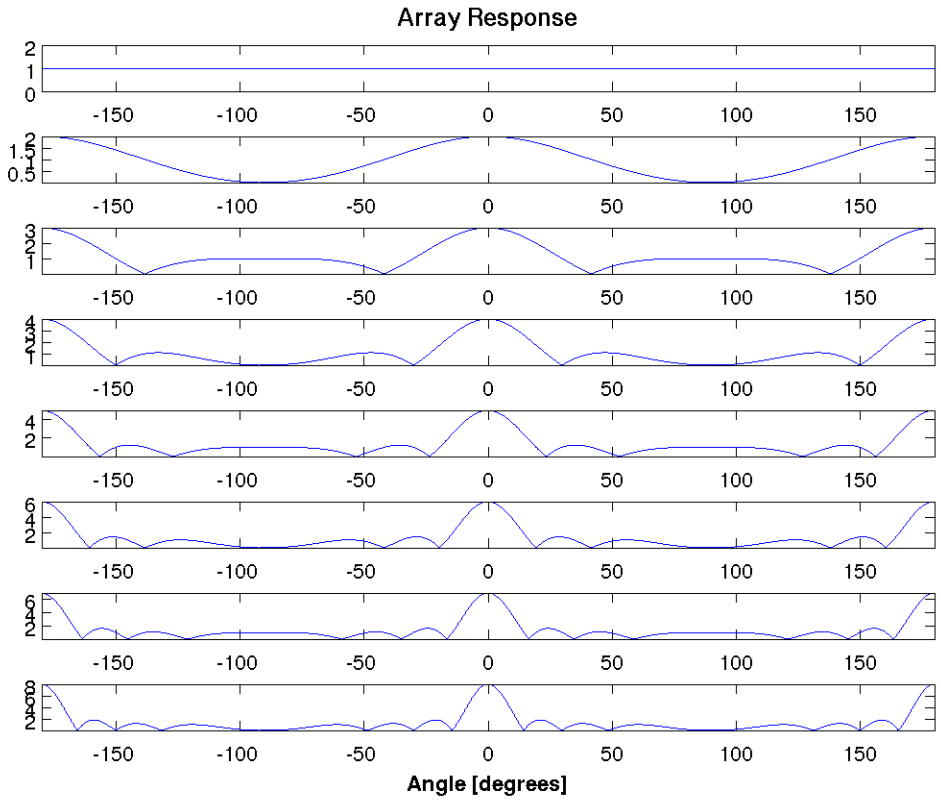


Figure 2.1: Array antenna pattern for a uniform linear array with $N = \{1, \ldots, 8\}$ (from top to bottom) elements spaced one-half wavelength apart. Each element has a radiation pattern of $S_i(\phi) = 1$.

The elements form a beam when constructive and destructive interference create maximums and minimums along different directions. We show the effect of adding elements to a linear array in Fig. 2.1.

If the signal is delayed in each element by some amount, the angle at which the electromagnetic waves add coherently changes. The literature commonly refers to this delay as adding a *phase taper* to the elements or giving them a *complex weight*. Phase tapers are chosen based on the distance waves must travel to add coherently. Fig. 2.2 illustrates the effect of inducing an incremental delay in the elements of a linear phased array to steer its beam in the desired pointing angle. We show the

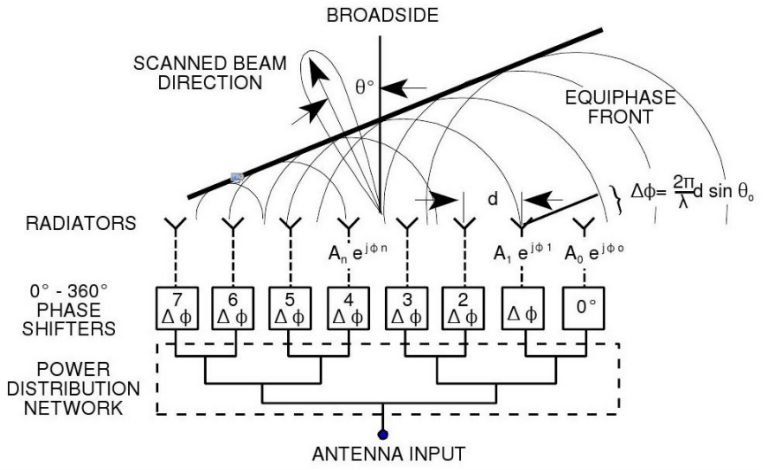


Figure 2.2: Illustration of beam steering in a centralized array antenna. An incremental delay induced in each element causes the summed response to interfere constructively in the desired direction [21].

beampattern of a linear phased array with different phase tapers that steer the beam's pointing direction in Fig. 2.3.

2.2.1 Array Factor and Beampattern

In antenna arrays, each antenna element has a pattern $S_e(\phi)$. If the phase in element i is Ψ_i, the array antenna's radiation pattern becomes [29]

$$S(\phi) = \sum_{i=1}^{N} S_e(\phi) e^{j\Psi_i} = S_e(\phi) \sum_{i=1}^{N} e^{j\Psi_i} = S_e(\phi) S_a(\phi) \tag{2.2}$$

where $S_a(\phi)$ is referred to as the *array factor*. The element antenna pattern $S_e(\phi)$ is assumed to be the same for all elements in an antenna array. The array factor characterizes the behavior of the array resulting from the array's geometry and the phase of the signal in its elements. If the elements that make up the antenna array are isotropic radiators, the element antenna pattern is $S_e(\phi) = 1$.

The literature commonly describes antenna patterns in terms of power, which we

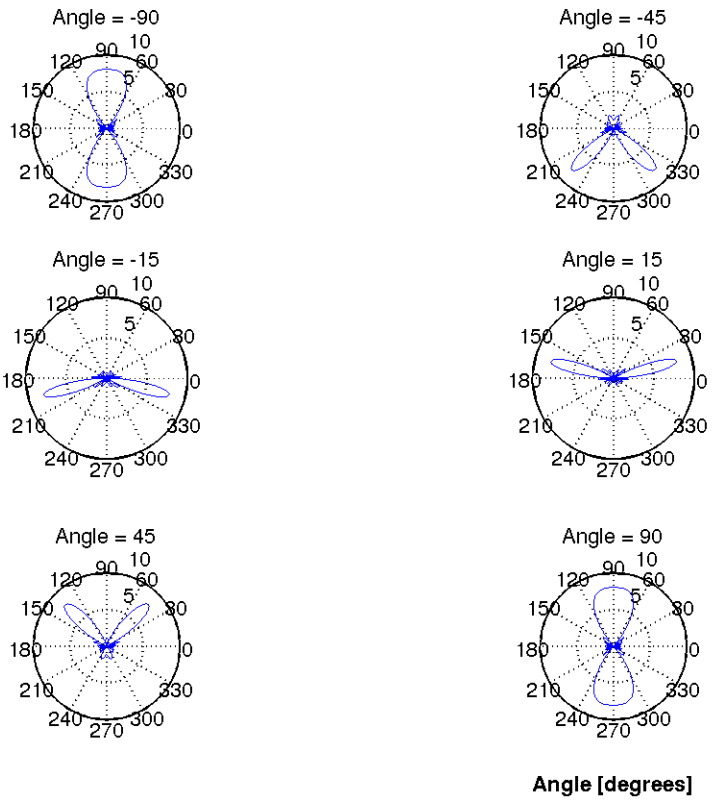


Figure 2.3: A uniform array antenna with $N = 8$ elements spaced 1/2 wavelength apart. The elements have the same geometry as the bottom plot in Fig. 2.1, but by setting the phase taper the beam can be steered in the desired pointing angle.

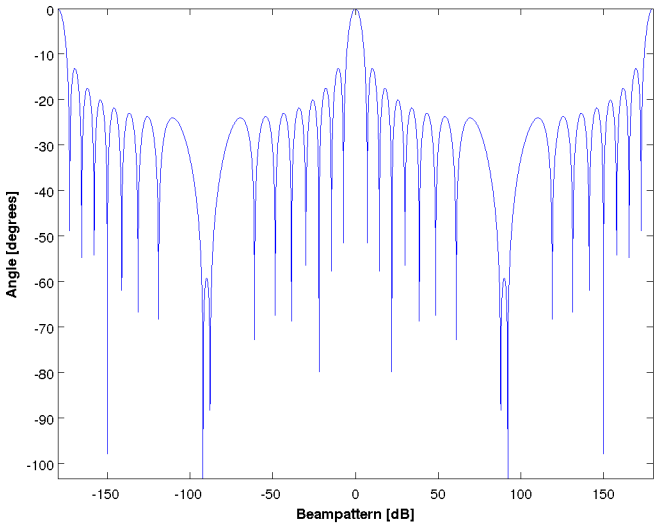


Figure 2.4: Beampattern of a uniform linear array with $N = 16$ elements spaced $d = \lambda/2$ apart.

refer to as their *beampattern*. The beampattern of an antenna array is found by [29]

$$P(\phi) = \frac{1}{N^2} \left| S(\phi) \right|^2 \tag{2.3}$$

and is typically expressed in dB. An example is shown in Fig. 2.4. The beampattern power allows us to measure how an antenna array transmits power in all directions.

2.3 Beam Pattern Analysis

Beampattern analysis comprises a large portion of this review, so we define some terminology associated with the beampattern of an antenna array. We will also review the structure of an antenna array beampattern and describe the scalar parameters beamwidth, directivity and gain; used to characterize a beampattern's shape and describe transmission performance.

The beampattern of an antenna array typically consists of a *main beam*, the maxi-

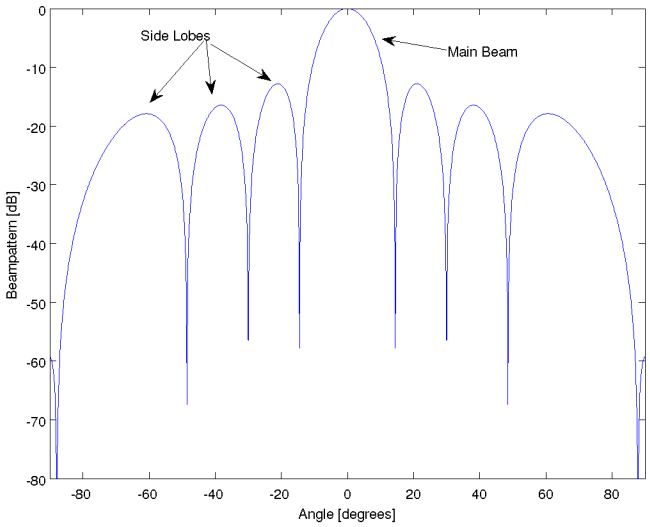


Figure 2.5: Beampattern of a uniform linear antenna array with $N = 8$ elements spaced 1/2 wavelength apart. The main beam is in the direction of maximum transmit, $\theta = 0$. Side lobes are maximums away from the pointing angle.

mum in the array antenna's pointing direction. Beampatterns also have *side lobes*, or maximums in the beampattern at directions away from the pointing angle, typically less than the main beam. Designers of antenna arrays are sometimes interested in minimizing the power of side lobes, which minimizes interference, but decreases the transmission efficiency [27]. Applying different complex weights to the elements of an antenna array affects the side lobe behavior. For example, binary weighted arrays [28] have zero side lobes, but increase the array antenna's beamwidth.

2.3.1 Beamwidth

Beamwidth allows us to define a scalar quantity that describes the beampattern shape. Often measured by the angular distance between the half-power level, we use the half-power beamwidth (HPBW) or 3dB-beamwidth extensively. Alternatively, the beamwidth could be measured by the distance between the first minima, referred

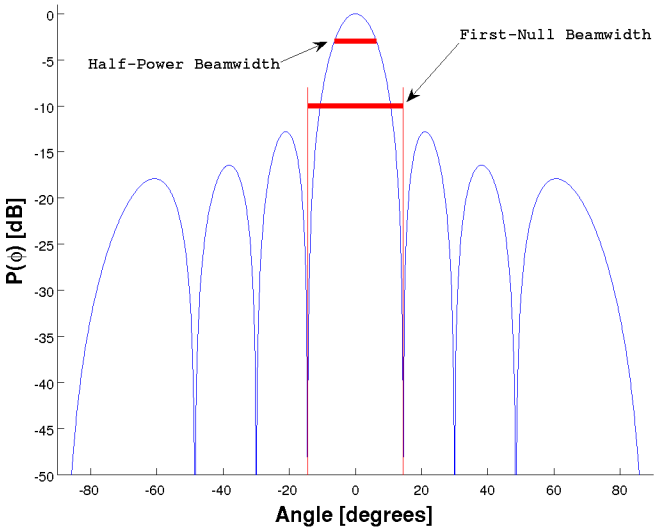


Figure 2.6: HPBW and FNBW for an antenna array with $N = 8$ elements spaced 1/2 wavelength apart, with reference to absolute power.

to as the first-null beamwidth (FNBW). Illustrations for each are shown in Fig. 2.6.

Beamwidth could help a designer determine how much error in the pointing direction can be tolerated. If the receiver requires at least half of the maximum possible receive power be available at its location, the pointing error must be less than HPBW/2.

2.3.2 Directivity and Gain

Directivity characterizes the amount of signal radiated along the pointing angle compared to the average power radiated in all directions. To compute the directivity of an antenna array in the direction ϕ, we use [8]

$$D(\phi) = \frac{P(\phi)}{\frac{1}{2\pi}\int_0^{2\pi} P(\varphi)d\varphi} \tag{2.4}$$

If we refer to directivity as a scalar value we are referring to its maximum directivity, or the directivity along the pointing direction $[D(\phi)]_{max} = D(\phi_{max})$.

The directivity is related to the gain of an antenna through its radiation efficiency η, where gain G is defined as [8]

$$G(\phi) = \eta D(\phi) \tag{2.5}$$

We use the gain to determine how much a signal is amplified in a given direction from the antenna array.

2.4 Transmission Analysis

During system design, it is useful to understand how the previously defined parameters affect the transmission of signals from the source to the destination. We use the array's gain and the Friis free-space path loss model [8] to determine the received power at any location.

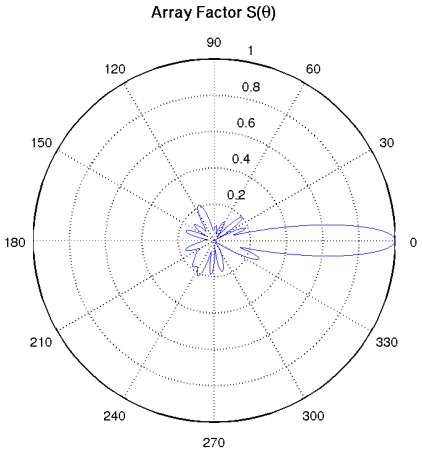
The input power P_{Tx} is the signal power generated from a single transmitting element, and the gain is computed from (2.4) and (2.5). If we assume the antenna array operates in an open area, devoid of objects to scatter and reflect its signal, the received signal power is [8]

$$P_{Rx}(\phi, d) = P_{Tx} G(\phi) \left(\frac{\lambda}{4\pi d} \right)^2 \tag{2.6}$$

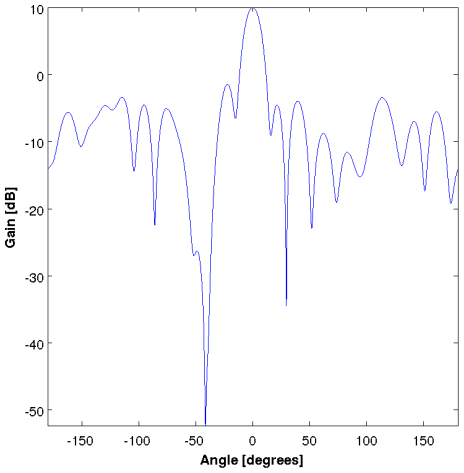
where d is the distance from the array and λ is the carrier wavelength.

2.4.1 Transmission Example

An example should help elucidate the topics introduced in this chapter and preview the work to done in the remainder of this review. We simulated an antenna



(a) Array factor $S(\theta)$ of a virtual antenna array with $N = 64$ elements randomly distributed over a $R = 200$ m circular area.



(b) Gain $G(\theta)$ of the system of distributed transmitters, computed from the array factor shown to the left.

array with $N = 64$ isotropic radiators randomly distributed over a disk with radius $R = 200$ m. A wavelength of $\lambda = 100$ m achieves preferable beam performance, for reasons discussed later in Chapter 5. The array factor was computed within our simulation framework and is shown in Fig. 2.7(a). From (2.4) and (2.5), and selecting a conservative value [3] of 55% for the radiation efficiency η, we computed the gain of the array, shown in Fig. 2.7(b)

If the transmitters are located in the center of a 100x100 km area, the distance and angle to each point can be computed. We then computed the received power using (2.6) and show the results in Fig. 2.7.

Each element transmits a power of 0.5 W, equal to the power transmitted by a cellular phone. We selected a conservative value [3] of 55% for radiation efficiency η. Within the main beam's coverage area, the received power is sufficient for communication purposes at distances up to 100 km.

Directions associated with sidelobes transmit enough power for eavesdroppers equipped with sensitive receivers. This motivates the work in [32], where the au-

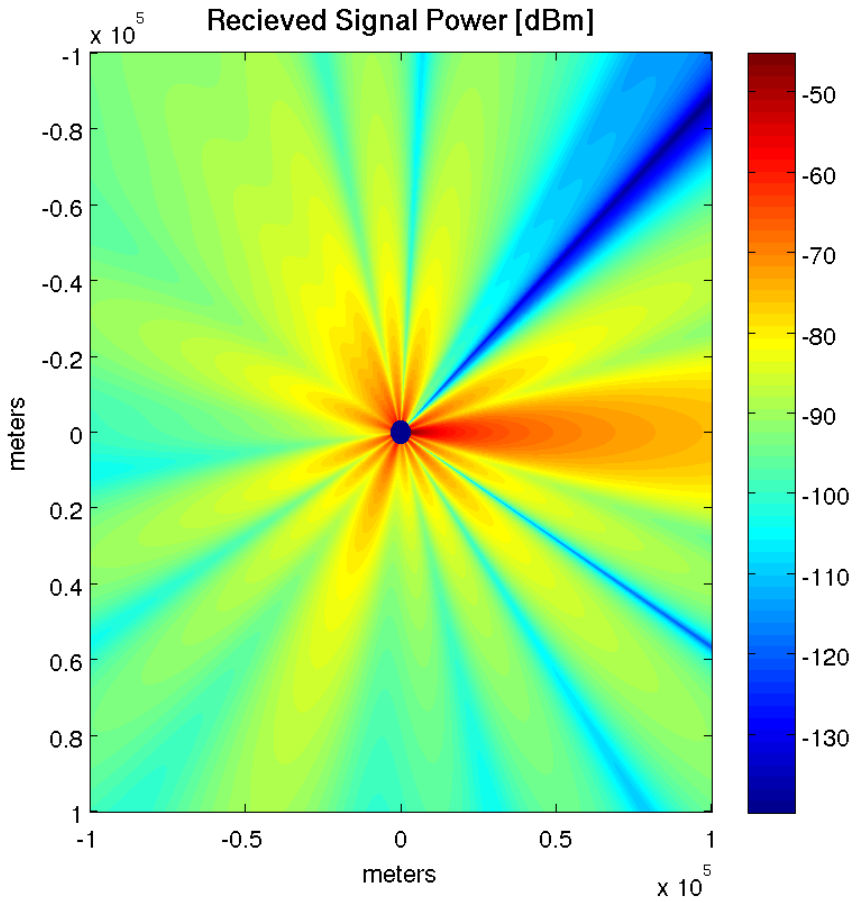


Figure 2.7: Received power at locations around the area the transmitters are distributed.

thors design a *nulling beamforer*, which selectively nulls the transmitted power in pre-defined directions through optimal weight selection.

2.5 Conclusion

Using the information provided in this chapter, we can describe the performance of antenna arrays. We have introduced how antenna arrays form a coherent beam and how it can be steered by inducing the proper phase delay in the signals through each element. We described the structure of a beampattern, and defined parameters used later in this review. We reviewed the transmission equation and used an example to show how an antenna array increases the received power along the pointing angle. We will use the information provided in this section to describe the performance of a system of distributed transmitters acting as a virtual antenna array.

3. Simulation Framework

3.1 Introduction

Studying a system of distributed transmitters acting as an antenna array requires the handling of a large number of parameters and calculations. To support research, we developed a simulation framework that handles this challenge and helps us gain a better understanding for the behavior of arrays under different configurations. We use the results of the simulation framework throughout this thesis. In this chapter, we review the simulation framework's features and demonstrate it in the simulation of linear arrays. The outputs are then verified by models found in literature.

Section 3.2 reviews the application's features, and how it simulates different arrays. Section 3.3 demonstrates the application in the simulation of linear arrays, and verifies its results with known models.

3.2 Application Features

The simulation framework was developed in MATLAB and includes a front-end GUI application shown in Fig. 3.1. The plotting window displays the beampattern of the simulated array, and the panels on the right configure its settings. The application display statistics for directivity and beamwidth on the lower panels.

The simulation starts by selecting the Run button in the Simulation panel. The framework computes the element locations from the array geometry settings provided. The signal phase for each element is computed to produce perfect coherence along the pointing angle. If errors are introduced into the system, they are applied prior to simulating the beampattern. The process is repeated multiple times if desired. Finally, the application plots the average beampattern and displays the directivity

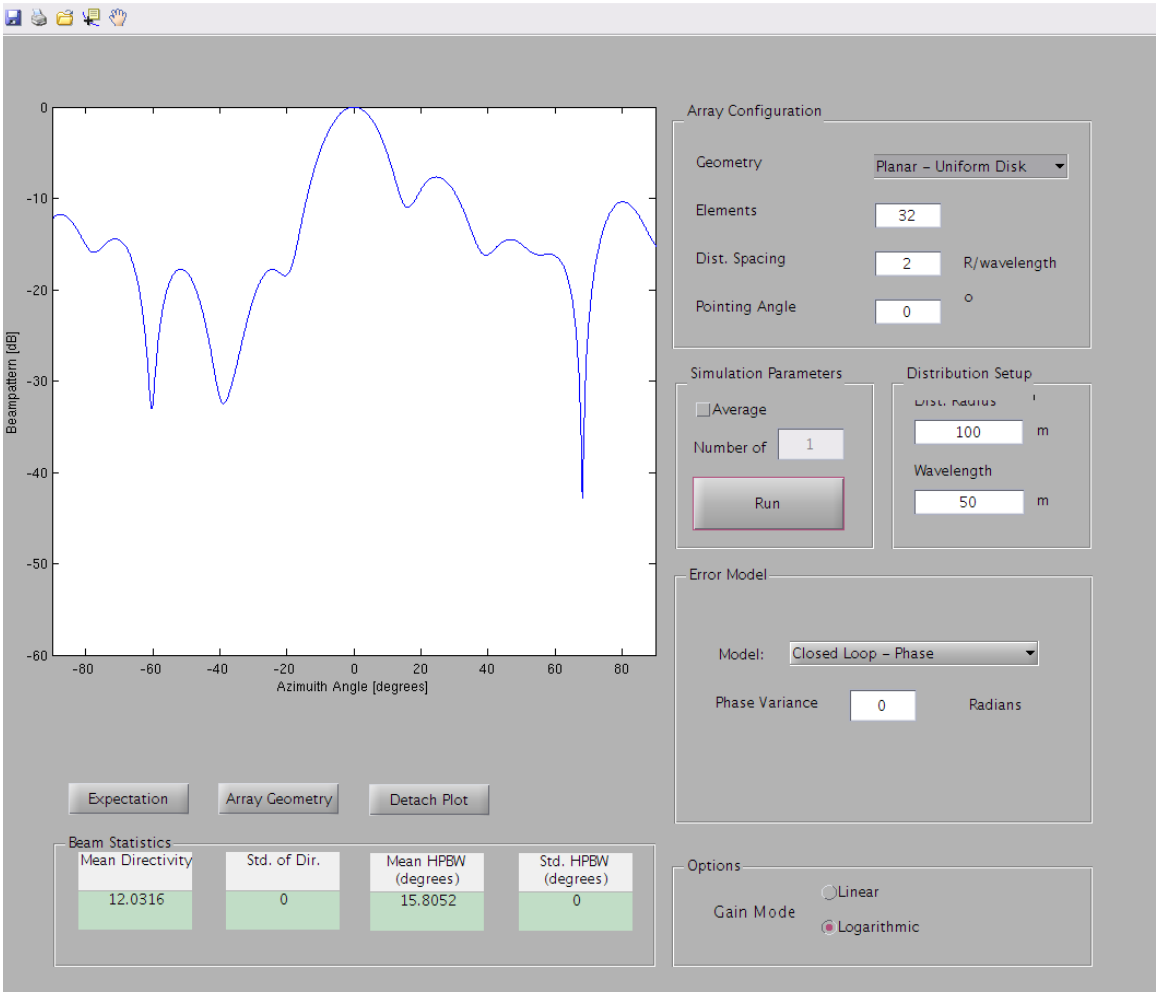


Figure 3.1: Current GUI for MATLAB application used to study array antennas.

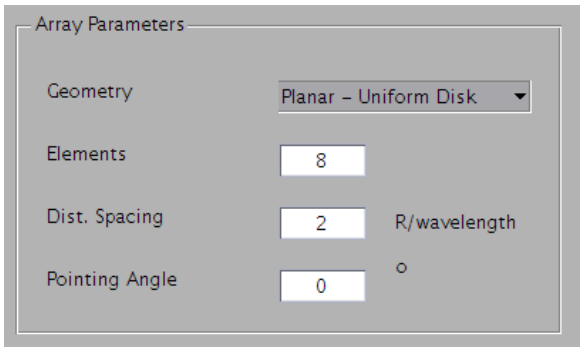


Figure 3.2: Array Parameter settings. The user can specify the number of elements, their spacing, and the pointing angle of the virtual antenna array.

and beamwidth statistics.

The framework uses the MATLAB Phased Array System Toolbox™ [22] to generate the array beampatterns. The toolbox requires the input of element positions, phase tapers, and the carrier frequency. Results from the toolbox are then used to numerically compute values for directivity and beamwidth.

3.2.1 Inputs

Settings for simulating different arrays are subdivided into four panels: Array Configuration, Distribution Setup, Error Model, and Simulation Parameters.

Array Parameters

The user can select either a linear or distributed array for simulation, along with the number of elements in the array, their spacing, and the pointing angle. The panel is shown in Fig. 3.2.

Simulation Parameters

The application can simulate one or multiple instances of an array, specified by this panel. The simulation starts by clicking the Run button located here.

Error Model

The user can specify a model for introducing errors into the array. For a linear array, the element positions are randomly perturbed. In a distributed array, random error can be introduced into the element's signal phase or its location by using the Closed-Loop or Open-Loop model, respectively.

3.2.2 Outputs

The GUI displays the average beampattern, and statistics for the directivity and beamwidth. The simulation framework saves each simulated beampattern, along with the directivity and beamwidth for each instance.

Models for expected beampatterns were incorporated into the application, and can be displayed alongside simulated beampatterns. The model for linear arrays with error was drawn from the work of Tseng and Cheng [23], which we use to verify the framework later in this chapter. The distributed model draws from the work of Ochiai et al. [17], and is the subject of review in Chapter 5.

The application can also display the array geometry. An example for each type is shown in Fig. 3.3.

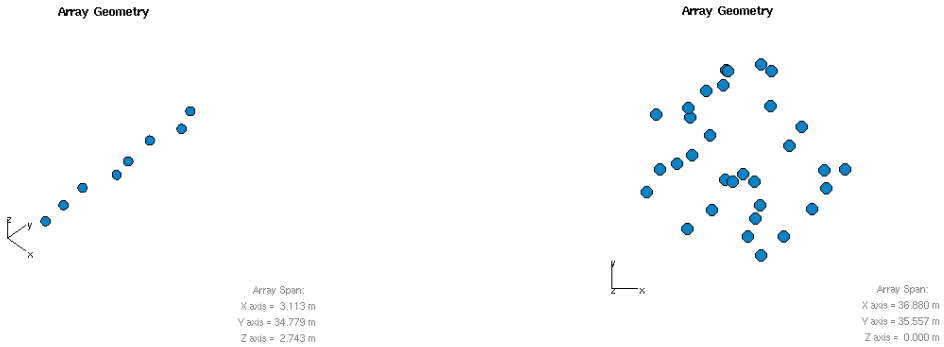
3.3 Verification

We sought to verify the application by comparing the beampattern of simulated linear arrays with models that describe them.

3.3.1 Linear Array

The beampattern of a linear array is well defined [29] by

$$P(\phi) = \frac{1}{N^2} \left| S(\phi) \right|^2 \tag{3.1}$$



(a) Linear (b) Distributed.

Figure 3.3: Examples of the instantaneos geometry of simulated arrays, generated using the Array Geometry button.

where

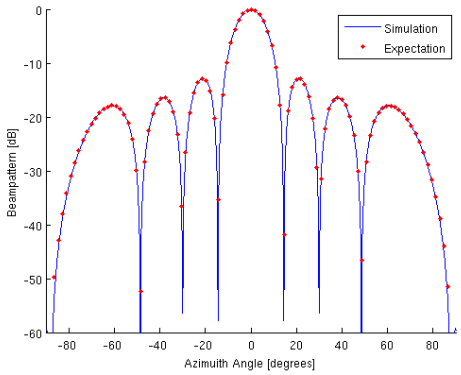
$$S(\phi) = \sum_{i=1}^{N} S_i(\phi) = \sum_{i=1}^{N} e^{jk_0(N-i)d[\sin\phi - \sin\phi_o]} \tag{3.2}$$

and N is the number of elements, $k_0 = \frac{2\pi}{\lambda}$ is the carrier wave number, ϕ is the azimuith angle, and ϕ_o is the pointing angle. Using this, we verify beampatterns generated using our simulation framework for a linear array.

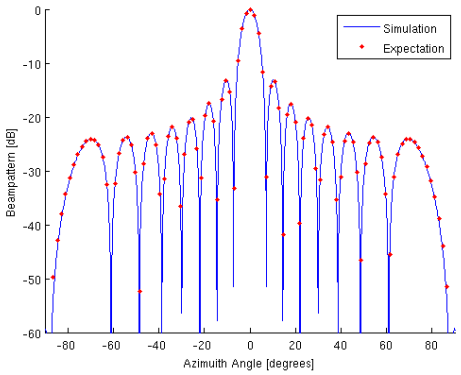
We simulated a single instance of different array configurations using the simulation framework. For a linear array, the framework computes element positions based on their spacing and carrier wavelength. The phase taper was calculated by

$$\psi_k = \frac{2\pi}{\lambda} dk \sin(\phi_0). \tag{3.3}$$

We vary the number of elements, their spacing and pointing angle. The result are shown in Figs. 3.4, 3.5, 3.6, alongside their expected results, found using (3.1) and (3.2). For all cases, the expected results match simulations generated using the framework.

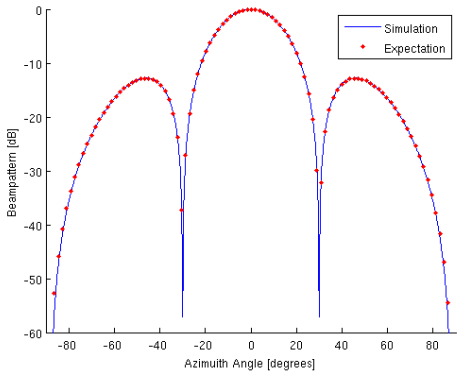


(a) $N = 8$ elements

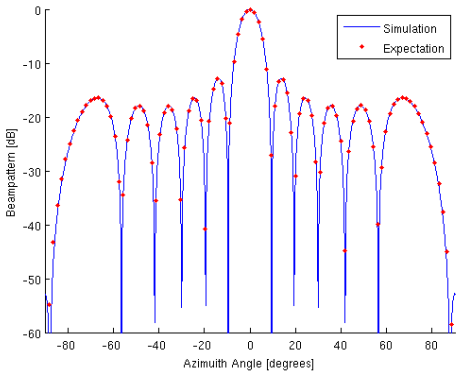


(b) $N = 16$ elements

Figure 3.4: Uniform linear array with spacing $d = \lambda/2$, and pointing angle $\phi_o = 0$.

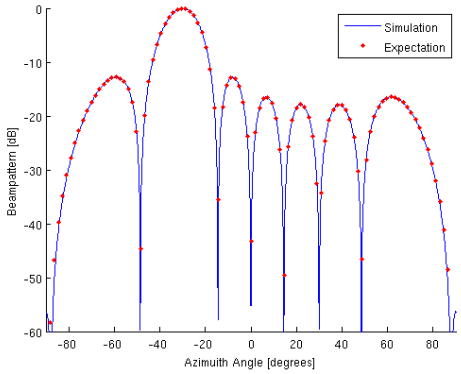


(a) $d = \lambda/4$.

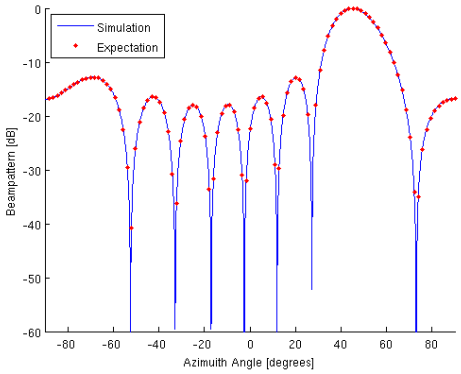


(b) $d = 3\lambda/4$.

Figure 3.5: Uniform linear array with $N = 8$ elements and $\phi_o = 0$ pointing angle.



(a) Uniform linear array with $\phi_0 = -30^o$.



(b) Uniform linear array with $\phi_0 = 45^o$.

Figure 3.6: Uniform linear array with $N = 8$ elements and $d = \lambda/2$.

3.3.2 Linear Array with Element Error

Much of this review focuses on the effect of element error on the beampattern of the array. In this section, we review how element position error affects the beampattern of a linear array. Tseng and Cheng [23] developed a model for the expected beampattern of linear arrays with random error in element position. We used Tseng and Cheng's findings to further verify our simulation framework.

Tseng and Cheng used an alternative definition for beampattern [23]

$$P(\mathbf{u}) = \sum_{m=1}^{N} \sum_{n=1}^{N} \exp\left\{j[(\psi_m - \psi_n) + k_0(\mathbf{r}_m - \mathbf{r}_n) \cdot \mathbf{u}]\right\} \tag{3.4}$$

where $\mathbf{u}$ is the unit directional vector, and ψ_n and $\mathbf{r}_n$ represent the phase and position of element n, respectively. Tseng and Cheng sought to find the expected power pattern $\hat{P}(\mathbf{u}) = E\left[P(u)\right]$ when the element positions were the vector random variables [23]

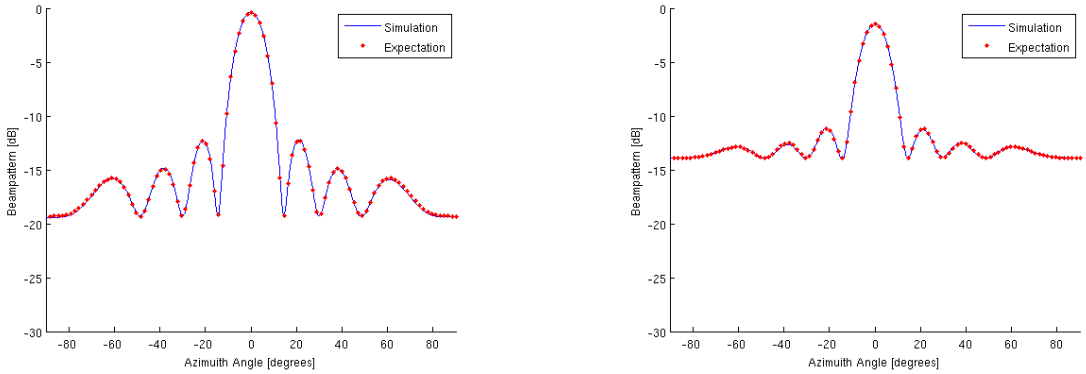
$$\mathbf{r}_n = \mathbf{r}_n^0 + \mathbf{l}_n \tag{3.5}$$

where $\mathbf{r}_n^0$ represents the mean value and $\mathbf{l}_n$ is the vector random variable representing position error. The expected power pattern becomes [23]

$$\hat{P}(\mathbf{u}) = \sum_{m=1}^{N} \sum_{n=1}^{N} \xi_{mn} \exp\left\{j[(\psi_m - \psi_n) + k(\mathbf{r}_m^0 - \mathbf{r}_n^0) \cdot \mathbf{u}]\right\}. \tag{3.6}$$

If location errors are uncorrelated gaussian random variables in three space, Tseng and Cheng found [23]

$$\xi_{mn} = \begin{cases} exp\left\{-(k_0\sigma)^2\right\}, & m \neq n \\ 1, & m = n \end{cases} \tag{3.7}$$



(a) $\sigma = 0.05\lambda$ (b) $\sigma = 0.10\lambda$

Figure 3.7: The average beampattern of a linear array with $N = 8$ element, $d = \lambda/2$ spacing and $\phi_0 = 0$ pointing angle. The element position were perturbed by a gaussian random variable with variance σ^2 in three space.

where σ^2 represents the error variance in element position.

Our simulation results match the findings of Tseng and Cheng, shown in Fig. 3.7. To compute the average simulated beampattern, we simulated 5000 instances of a linear array with element positions represented by the random variable in (3.5).

3.4 Conclusions

We have verified our simulation framework by comparing the simulated beampattern of linear arrays with expected results. Our simulation framework allows the user to change array configurations and view its performance through its beampattern, directivity and beamwidth. In Chapter 5, we use the framework to study arrays with random geometry. We also reuse the framework for generating element errors, and extend it to study errors introduced by the synchronization of transmitters detached in space, the topic of our next chapter.

4. Synchronization Methods

4.1 Introduction

To beamform using distributed transmitters requires each node has the same carrier frequency and a properly controlled signal phase. Conventional arrays use a feed network to connect array elements to the signal source, which ensures each element transmits a version of the same signal. Because our transmitters are detached in space, other methods must be used to create a *virtual feed network*. We review methods researched to accomplish the task of carrier frequency synchronization and signal phase control. We also review a quantitative measure of phase error in the nodes. Finally, we discuss prototypes developed and some recent trends in research.

Section 4.2 provides an overview of researched synchronization methods. We review working prototypes in Section 4.3, and then discuss the recent advances and future directions of research in Section 4.4. We summarize the methods and discuss their implications in Section 4.5.

4.2 Overview of Methods

In evaluating synchronization methods we must consider the following: the speed at which phase and frequency converge to the desired values, the processing complexity at the sources and destination, the scalability as the number of transmitters grows, and the overhead time required. Each objective cannot be simultaneously satisfied, so we consider the tradeoffs and intended application when deciding the best solution for a particular situation.

It should be mentioned that the nodes must achieve some level of timing synchronization in order to transmit the proper signal at the proper time. For relatively low

data rates (∼100 kb/s), the required level of timing synchronization can be achieved using well-known algorithms like RBS [6].

There are two methodologies for achieving carrier synchronization and phase control, which differ by the amount of interaction between the sources and destination.

- Closed-loop synchronization relies on input from the destination to control frequency and phase while source interactions are minimal.
- Open-loop synchronization methods leverage more source interactions, with minimal input from the destination.

4.2.1 Closed-Loop Full-Feedback

One of the first methods developed to synchronize transmitters to act as a virtual antenna array was studied by Tu and Pottie in 2002 [24]. Later named closed-loop full-feedback, the method uses feedback to control the phase of each individual node. The process occurs as follows:

1. The destination broadcasts a beacon signal to be received by the source nodes.
2. The sources nodes 'bounce' the beacon signal back to the destination, which estimates the propagation delay. A multiple access scheme is used to discriminate signals from each source. The original authors suggest modulating the return signal using DS-CDMA.
3. Based on the propagation delays, the destination calculates the required phase at each node for coherent transmission. The information is transmitted back to the source nodes.
4. Each source adjusts its signal phase based on the feedback information, and the nodes transmit as a collaborative beamformer.

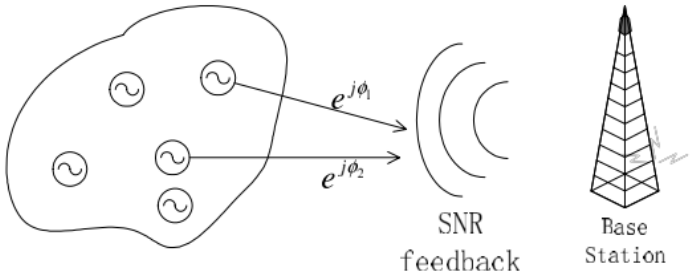


Figure 4.1: Phase synchronization using one-bit feedback [12]

The nodes converge to the correct signal phase after one iteration, assuming the channel does not change throughout the process. The method requires little work from the source nodes, which rely on the destination to compute all their signal phases, a practice that may not scale well as the number of sources grows. Tu and Pottie also discovered there was an optimal amount of feedback for balancing beamforming performance with information throughput.

A potential drawback in using this method is that the destination must detect the signal sent from each source node. This is problematic because in some applications collaborative beamforming is used so the *combined* transmission has enough power to be detected at the destination.

4.2.2 Closed-Loop One-Bit Feedback

This next method uses beamforming performance as feedback information for source synchronization. Closed-loop one-bit feedback [12], depicted in Fig. 4.1, occurs as follows:

1. Each source *randomly* adjusts the phase of its signal.

2. The sources transmit as a collaborative beamformer.

3. The destination estimates the SNR of the received signal and determines whether it has improved. If the SNR has improved, the sources keep their randomly selected phases. Otherwise, the sources revert back to their original phases.

This forms one iteration of the procedure, and the process continues until the signal phases converge. However, the process takes time. If there are N nodes, it takes roughly $5N$ iterations to achieve 75% the ideal beamforming amplitude gains. The process scales well, as convergence time increases linearly with N [12].

The process is relatively simple, making it a practical solution for a realizable system. There exists a good balance between processing done at the source and destination. A prototype utilizing this method was developed [15], which we review later in the chapter.

A potential drawback in using this system is its slow convergence time. Some alternatives have been proposed, which use more feedback bits, resulting in faster convergence [25, 26].

This scheme accounts for the nodes' total phase offset, which includes local oscillator drift. The process still requires an initial carrier frequency synchronization. A similar feedback method explicitly accounts for carrier frequency synchronization [16].

4.2.3 Open-Loop Master-Slave

A key drawback in using the previous methods are the high costs associated with long-range communications. To take advantage of the lower communication cost at short range, open-loop methods rely on more interactions between the source nodes. Open-loop methods also alleviate processing costs incurred at the destination associated with closed-loop methods.

The open-loop master-slave method, first proposed by Mudumbai, Barriac and Madhow in 2007 [11], was inspired by master-slave frequency synchronization. A

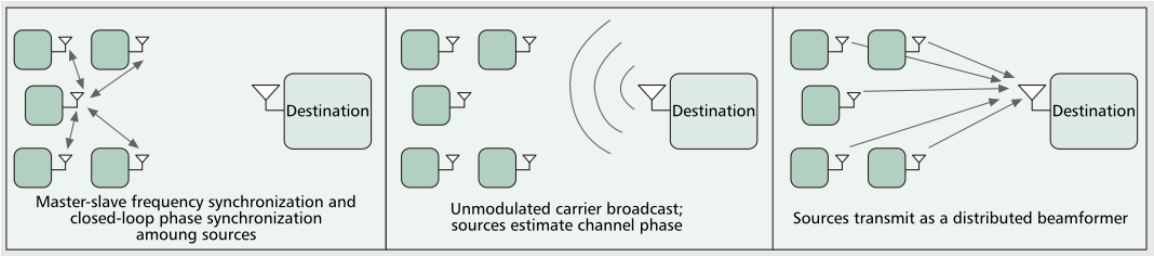


Figure 4.2: Master-slave open-loop synchronization process [13]

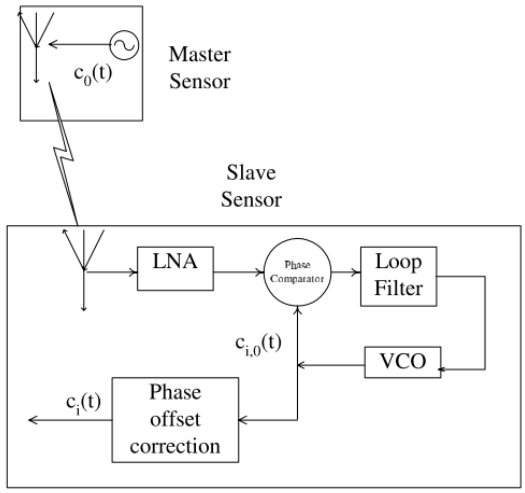


Figure 4.3: Master-slave synchronization [11].

master node is designated, which facilitates the carrier frequency synchronization of slave nodes. The process is depicted in Fig. 4.2 and outlined as follows:

1. The master node broadcasts a beacon signal at the reference frequency.
2. The slave nodes synchronize their carrier frequency using the received beacon. Mudumbai et al. propose the use of an analog PLL circuit, shown in Fig. 4.3, for carrier synchronization.
3. The destination broadcasts a beacon, which the source nodes use to estimate the complex channel response between them and the destination.

4. The nodes set their transmit gain to the complex-conjugate of the channel response. This effectively applies the proper phase delay for collaborative beamforming.

5. The sources transmit as a collaborative beamformer.

The process repeats for continual synchronization of the nodes' phase and carrier frequency. Mudumbai et al. propose a time division duplex mode of operation, shown in Fig. 4.5. The processes are divided into two timeslots: one for carrier synchronization, and another for transmission. The transmission timeslot includes channel estimation and collaborative beamforming. A system designer would like to maximize the time spent beamforming in order to maximize information goodput.

The open-loop master-slave method reduces the number of interactions between the sources and destination, compared with closed-loop methods. The process also scales up well, as each node is responsible for synchronizing its own frequency and phase. However, the additional work placed on the nodes must be accounted for by increasing their processing power. The process also incurs a phase error in the nodes that changes over time.

Evaluation of Error

To study realized beamforming gains, it is useful to determine the phase error in the nodes throughout the synchronization process. Phase error is governed by the durations of the *sync* and *transmit* timeslot. During the *sync* timeslot, the slave nodes receive the master's beacon signal, and syncrhonize their phase and frequency using a second-order PLL circuit, like the one shown in Fig. 4.4. The PLL has a closed-loop transfer function [10]

$$H(s) = \frac{s^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{4.1}$$

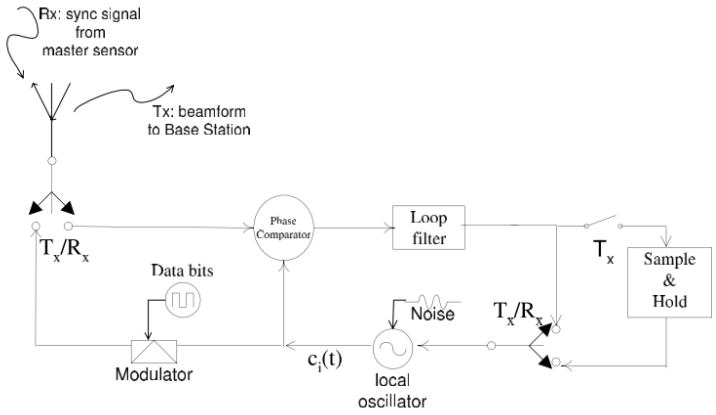


Figure 4.4: Transciever schematic [11].

where ω_n is the natural frequency and ξ is the damping ratio. To reduce overshoot, $\xi = 1$. $T_s \approx \frac{4}{\omega_n}$ gives us a measure for settling time, which must be less than synchronization timeslot duration. The loop's low-pass bandwidth is approximately ω_n, which contributes to the internal phase noise introduced by the transceivers local oscillator. There exists a tradeoff between minimizing the *sync* timeslot and minimizing phase noise.

To quantitatively study phase noise, the phase noise was modelled as a cyclostationary random process with period $T = T_1 + T_2$, where T_1 is the *sync* timeslot duration and T_2 is the *transmit* timeslot duration. Phase noise is considered a zero-mean gaussian random process, described by its variance at points A, B, and C along the synchronization frame shown in Fig. 4.5. Random phase offsets are denoted ϕ_A, ϕ_B, and ϕ_C, with standard deviations σ_A, σ_B, and σ_C, respectively.

At time instant B, after the *sync* timeslot, the phase offset converges to a fraction ρ of its initial value ϕ_A, including a noise term [11]

$$\phi_B = \rho\phi_A + \psi_1 \tag{4.2}$$

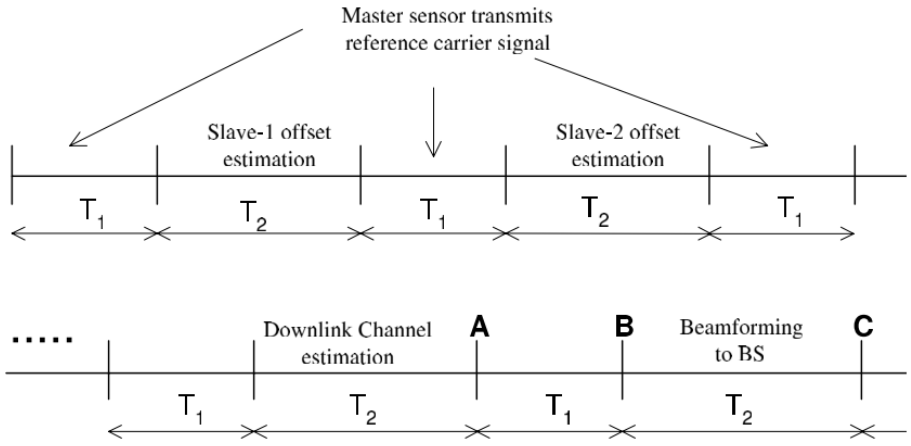


Figure 4.5: Time division duplex frame [11].

ϕ_B is small by design, and has a variance of [11]

$$\sigma_B^2 = \rho^2 \sigma_A^2 + N_p \omega_n \tag{4.3}$$

where N_p is the local oscillator's internal noise spectral density and ω_n is the loop's bandwidth.

Also at time instant B, the frequency is sampled for the *transmit* timeslot, and there exists a small random offset Δf from the reference frequency, including a noise term [11]:

$$\Delta f = \rho \omega_n \phi_A + \omega_n \psi_3 \tag{4.4}$$

with variance [11]

$$\sigma_f^2 = \rho_2 \omega_n^2 \phi_A^2 + \omega_n^3 N_p. \tag{4.5}$$

The frequency offset at the beginning of the *transmit* timeslot translates into a linearly increasing phase offset, giving [11]

$$\phi_C = \Delta f T_2 + \phi_B + \psi_2. \tag{4.6}$$

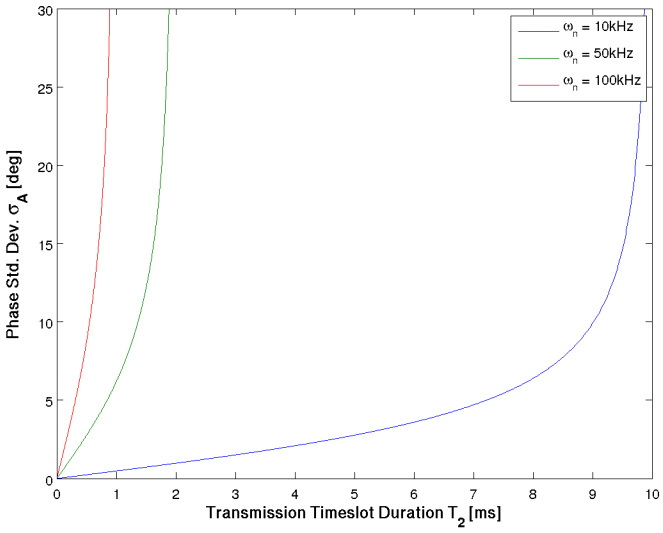


Figure 4.6: Phase standard deviation as a function of timeslot duration, from (4.8) where $N_p = -101$ dBc/Hz and $\rho = 1\%$.

Again, ϕ_B is small by design, and ψ_2 is a stationary term, so both can be safely neglected compared with the term describing linear drift, leaving [11]

$$\sigma_C^2 \equiv \sigma_A^2 = \sigma_f^2 T_2^2. \tag{4.7}$$

Combining (4.5) and (4.7) gives us a measure for the maximum phase variance that occurs after the *transmit* timeslot, before the *sync* timeslot [11]:

$$\sigma_A^2 = \frac{N_p \omega_n^3 T_2^2}{1 - \rho^2 \omega_n^2 T_2^2} \tag{4.8}$$

Notice that the phase variance increases without bound as $T_2 \to \frac{1}{\rho\omega_n}$. We show T_2 for several values of ω_n in Fig. 4.6. If we set the transmission timeslot duration to a constant fraction of the total frame duration, phase variance decreases as we increase frame duration. However, increasing frame duration may introduce latency issues.

We now have a model for describing the nodes' phase variance with respect to the

synchronization process. By changing the timeslot durations, we control the phase error experienced in the nodes.

4.3 Prototypes

Several prototypes have been developed in recent years that demonstrate the feasibility of these systems and discover nonidealities inherent in real devices.

The first prototype was developed in 2006 to demonstrate the one-bit feedback method [15]. The prototype consisted of three transmitters, synchronized using a common clock. The prototype was able to achieve 90 percent of ideal beamforming power gains after approximately 60 iterations.

Recently, distributed beamforming has been demonstrated on software-defined radios, and leverage digital signal processing for synchronization [19].

4.4 Recent Advances and Future Directions

Recent trends show an interest in the use of digital signal processing to do distributed beamforming [14]. Most use a Kalman filtering approach for carrier and phase synchronization. The Cramer-Rao lower bound for phase and frequency variance was computed, and shows significant suboptimality in using analog PLLs, motivating the use of optimal digital processing to improve performance [20].

It should also be mentioned that true open-loop syncrhonization could be achieved without any input from the destination if we knew the nodes' location and the direction of transmit.

4.5 Conclusion

We have reviewed several methods developed to syncrhonize transmitters for collaborative beamforming. Closed-loop methods rely heavily on input from the des-

tination, while open-loop methods utilize more source interactions. The open-loop master-slave method first synchronizes the nodes' carrier frequency, then uses a beacon signal from the destination to compute channel estimates and do collaborative beamforming. We reviewed the phase variance in nodes using the open-loop master-slave method for synchronization, and will test its effect on collaborative beamforming performance in the next chapter. Finally, some recent prototypes were reviewed that demonstrate collaborative beamforming using real hardware and the latest digital processing techniques.

5. Beamforming Performance Analysis

5.1 Introduction

In wireless sensor networks, nodes are free to take any configuration. We review the beamforming performance of a virtual array consisting of the nodes' transmitters. Since wireless sensor networks are not constrained in geometry, we study the beamforming *statistics* of randomly distributed array elements. We found that a good beam can still be formed using an array of randomly placed elements, and that changing the nodes' distribution changes their beamforming behavior. We review the beampattern shape, and the statistics of parameters that describe it. We also review models for the beampattern distribution, which provide a means for quantitatively measuring the beam's performance. Using the information presented, we hope to later show the design a system with desirable beamforming properties.

The first study of random arrays occurred within the antenna design community in the 1960s. Lo [9] showed that a good beam can be formed using a large number of randomly spaced elements in a linear configuration. This chapter reviews the work of Ochiai, Mitran, Poor, and Tarokh [17], who extend Lo's study to virtual arrays in two dimensions. Ochiai et al. studied the problem within the framework for wireless ad hoc sensor networks and published results for the beamforming statistics, which we review using the simulation framework presented in Chapter 3.

The chapter is organized as follows. Section 5.2 provides a description of the system model and reviews the beampattern of a random virtual array. Section 5.3 reviews the average beamforming properties of random arrays, including beampattern, beamwidth and directivity. We review statistical the distribution of the beampattern in Section 5.4 and Section 5.5 reviews a bound on beampattern sidelobe power. The

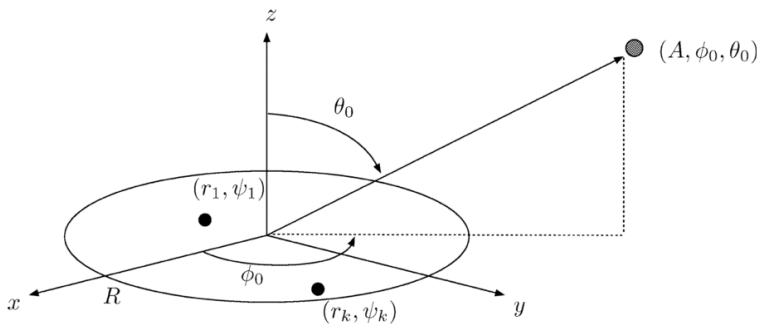


Figure 5.1: Beamforming system geometry [17].

effect of element errors are reviewed in Section 5.6 and we summarize our findings and discuss their implications in Section 5.7.

5.2 System Model and Beampattern

The geometry and notation for a system of distributed transmitters is shown in Fig. 5.1. All transmitters are located in the x-y plane, with locations specified by the polar coordinates (r_k, ψ_k). The destination's location is specified by the spherical coordinates (A, ϕ_0, θ_0). Following standard antenna conventions [8], the angle of transmission is represented by the elevation angle $\theta \in [0, \pi]$ and the azimuth angle $\phi \in [-\pi, \pi]$.

In our review of beampattern performance, a number of assumptions were made to simplify analysis [17].

1. The nodes and channel are static throughout the communication period.

2. Each node already has the message to be transmitted, so each node sends a version of the same signal.

3. Node locations are random, following a uniform distribution over a two-dimensional disk with radius R. This assumption is sufficient when transmitters are deployed

in a constrained area, like a field, indoors, or on top of buildings. Ahmed [1] evaluates the beamforming performance when node locations are gaussian distributed. Research [2, 7, 18] shows that gaussian distributed nodes are more likely when dropped from an airplane to cover a large area. Ahmed found that the beam of gaussian distributed nodes behave qualitatively similar to those studied by Ochiai et al.

4. Each transmitter is equipped with a single isotopic antenna.

5. All nodes transmit identical energies, and the path loss between each transmitter and the destination is equal, so the underlying model follows the framework for a phased array antenna.

6. There is no refection or scattering of the signal, thus there is no multipath fading or shadowing. In [31], the authors propose a distributed beamforming algorithm that compensates for the effects of local scattering typical of urban or suburban environments.

7. The nodes are sufficiently separated so that mutual coupling effects are negligible and ignored.

If the nodes are treated as a virtual antenna array, the array factor is computed using the signal phase of each node. To steer the beam, the signal phase Ψ_k of node k is set to [17]

$$\Psi_k = -\frac{2\pi}{\lambda} d_k(A, \phi_0, \theta_0) \tag{5.1}$$

where λ is the carrier wavelength and $d_k(\phi, \theta)$ is the euclidean distance to reference location (A, ϕ, θ).

Given a realization of random node locations $\mathbf{r} = [r_1, r_2, \ldots, r_N] \in [0, R]^N$ and

$\psi = [\psi_1, \psi_2, \ldots, \psi_N] \in [-\pi, \pi]^N$, the array factor is [17]

$$F(\phi, \theta|\mathbf{r}, \psi) = \frac{1}{N} \sum_{k=1}^{N} e^{j\Psi_k} e^{j\frac{2\pi}{\lambda} d_k(\phi,\theta)} \tag{5.2}$$

where N is the number of nodes.

The far-field distance can be approximated as $d_k(\phi, \theta) \approx A - r_k \sin(\theta)\cos(\phi - \psi_k)$ [17], giving an alternative method for element phasing. The phase of each node can be set to [17]

$$\Psi_k^{\dagger} = \frac{2\pi}{\lambda} r_k \sin(\theta_0)\cos(\phi_0 - \psi_k) \tag{5.3}$$

which gives the array factor [17]

$$\tilde{F}(\phi, \theta|\mathbf{r}, \psi) = \frac{1}{N} \sum_{k=1}^{N} e^{j\Psi_k^{\dagger}} e^{j\frac{2\pi}{\lambda} d_k(\phi,\theta)} \tag{5.4}$$

The array factor described by (5.2) and (5.4) are equivalent, providing two ways to form a beam. Setting the element signal phase using (5.1) requires knowledge of the distance to the destination. In closed-loop synchronization methods, the channel estimate or propogation delay effectively measures this distance. True open-loop element synchronization requires the transmitters know their location and the direction of transmit, so (5.3) would be used to compute element signal phase.

To simplify analysis, we only consider a destination that lies in the same plane as the array elements, thus $\theta_0 = \pi/2$. Since the node distribution is symmetric, the azimuth pointing angle ϕ_0 does not alter the results, so we set $\phi_0 = 0$.

By defining the compound random variable $z = \frac{r_k}{R}\sin(\psi_k - \phi/2)$, Ochiai et al. write the far-field array factor as [17]

$$F(\phi|\mathbf{z}) = \frac{1}{N} \sum_{k=1}^{N} e^{-j4\pi\tilde{R}\sin(\frac{\phi}{2})z_k} \tag{5.5}$$

where $\tilde{R} = R/\lambda$ represents the node distribution spread normalized by the carrier wavelength.

Finally, the far-field beampattern is [17]

$$P(\phi|\mathbf{z}) \triangleq \left|\tilde{F}(\phi|\mathbf{z})\right|^2 \tag{5.6}$$

$$= \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N} e^{-j\alpha(\phi)(z_k - z_l)} \tag{5.7}$$

where $\alpha(\phi) \triangleq 4\pi\tilde{R}\sin\frac{\phi}{2}$.

5.3 Average Beam Properties

The average beam properties are useful in determining the behavior of a random array in relation its element distribution. We review the average beampattern, beamwidth and directivity to determine how node configuration affects beamforming behavior.

5.3.1 Average Beampattern

Ochiai et al. found the average beampattern P_{av} by taking the expectation of (5.7) over all realizations of $\mathbf{z}$ [17].

$$P_{\text{av}}(\phi) \triangleq E_{\mathbf{z}}\left\{P(\phi|\mathbf{z})\right\} = \frac{1}{N} + \left(1 - \frac{1}{N}\right)\left|2 \cdot \frac{J_1(\alpha(\phi))}{\alpha(\phi)}\right|^2 \tag{5.8}$$

where $J_n(\cdot)$ is the n-th order Bessel function of the first kind.

The first term in (5.8) represents a constant beampattern power in the sidelobe region, inversely related to the number of transmitters. To verify (5.8), we averaged the beampatterns of 5000 random array realizations. Fig. 5.2 shows the comparison, which reveals that the original authors' expression for average beampattern was correct. By visual inspection, we see that beampattern shape depends on the num-

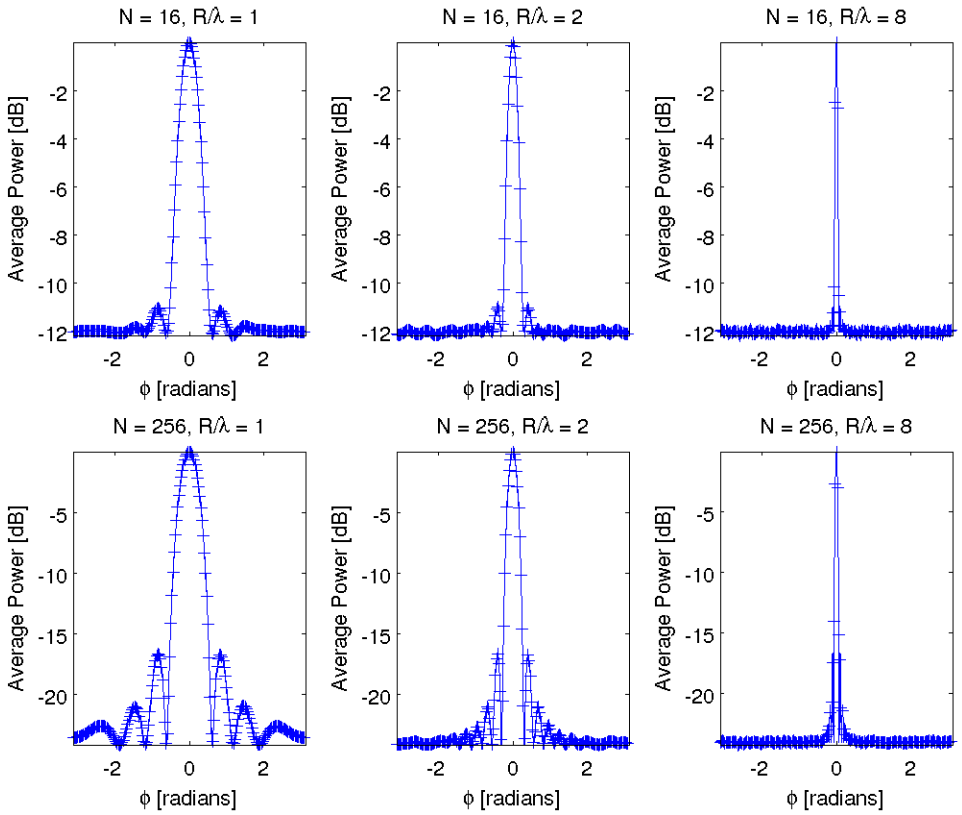


Figure 5.2: Average beampattern of random arrays. The solid lines represent (5.8) and the plus-signs show the average simulated beampattern of 5000 random array instances.

ber of nodes and their distribution. Increasing $\tilde{R}$ results in a narrower beam, while increasing N decreases the sidelobe power level.

5.3.2 Three-dB Beamwidth of Average Beampattern

A crucial parameter of the beampattern is its beamwidth. A narrow beam provides the most efficient use of power, but precise knowledge of the destination's location. A wider beam provides greater stability against errors in the array's pointing angle.

By numerically solving $P_{av}(\phi_{\text{av}}^{\text{3dB}}) = 1/2$, Ochiai et al. found the 3-dB beamwidth

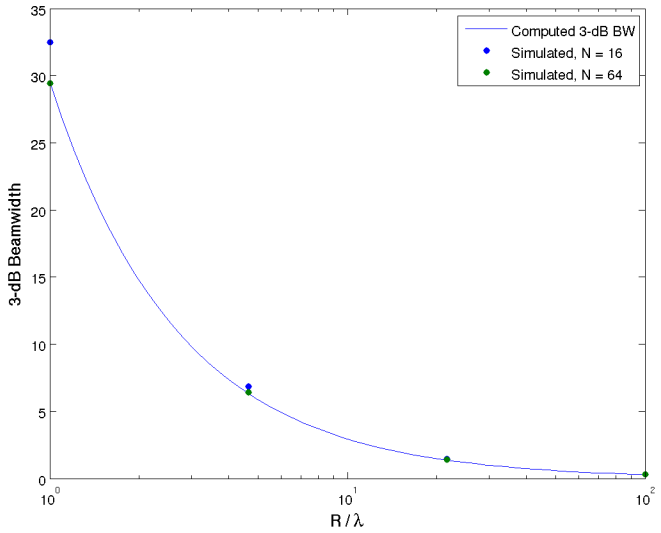


Figure 5.3: Theortical and simulated average 3-dB beamwidths for various $\tilde{R}$ and N

to be [17]

$$\phi_{\text{av}}^{\text{3dB}} = 4 \arcsin\left(\frac{0.1268}{\tilde{R}}\right). \tag{5.9}$$

As expected, the average beamwidth depends only on the spread of transmitting nodes $\tilde{R}$. Theoretical and simulated values for $\phi_{\text{av}}^{\text{3dB}}$ are shown in Fig. 5.3, and closely align at high values of $\tilde{R}$, regardless of the number of transmitters N. To decrease beamwidth, a designer must either increase the area over which the nodes are spread, or decrease their carrier wavelength.

5.3.3 Average Directivity

Directivity measures the concentration of signal radiated in the direction of interest, providing a measure of the gain achieved by the node configuration. Measuring the directivity per transmitter provides a measure for the configuration's beamforming efficiency.

Ochiai et al. derived an expression for the average directivity of a random array,

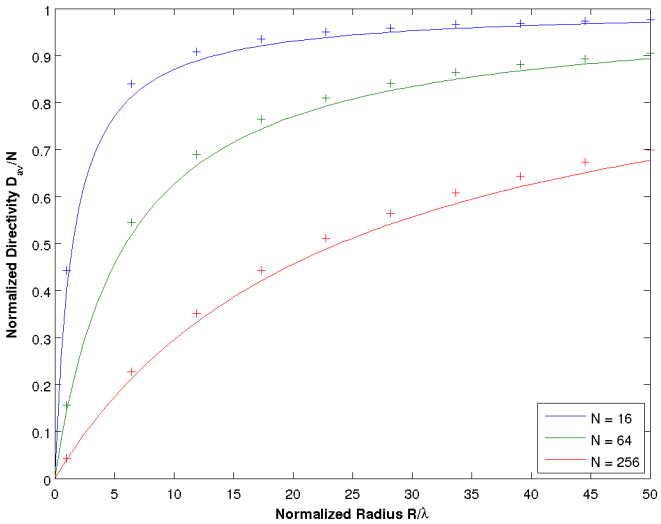


Figure 5.4: Normalized directivity versus effective distribution radius. Plus signs represent the average directivity of 5000 random array realizations and the solid line represents the lower bound for average directivity

and claimed that directivity approaches N as $\tilde{R} \to \infty$. The bound is illustrated in Fig. 5.4, where we show the average directivity of 5000 random arrays for each point, along with the analytical bound. Increasing the area of node distribution increases the directivity per transmitter, however increasing the number of nodes in the same area decreases the array's efficiency.

The following bound, derived by Ochiai et al., shows the directivity per node D/N is inversely related to the one-dimensional node density $N/\tilde{R}$ [17]:

$$\frac{D_{\text{av}}}{N} \geq \frac{1}{1+\mu\frac{N}{\tilde{R}}} \tag{5.10}$$

where μ is a positive constant, independent of system parameters ($\mu \approx 0.09332$). The results of another simulation are shown in Fig. 5.5, which verify the bound in (5.10).

To maximize efficiency, the elements must be distributed as sparsely as possible. However, sparsely distributed elements form a narrow beam, a tradeoff that must be

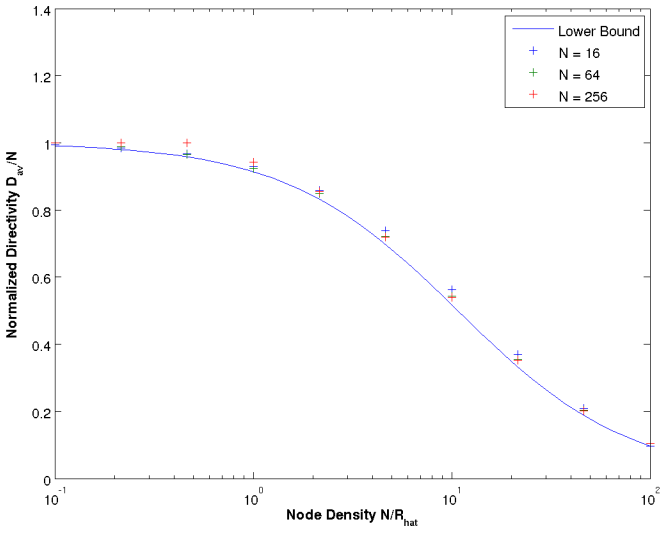


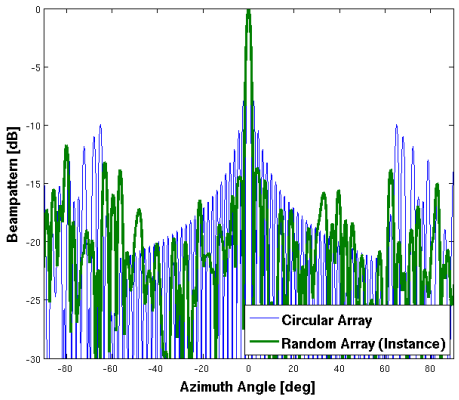
Figure 5.5: Lower bound on normalized directivity (5.10) alongside simulation results for $N = 16, 64, 256$ elements.

considered in the system design.

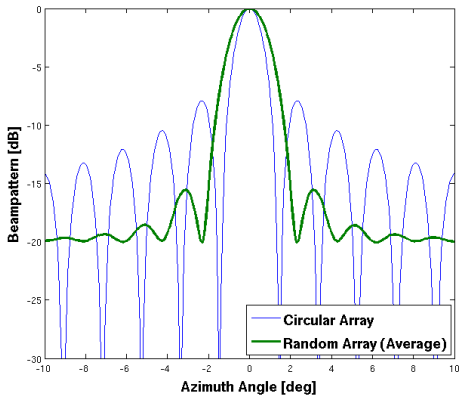
5.3.4 Comparison with Circular Antenna Array

We would like to quantify the effect of random positions on the beampattern of a virtual array against a conventional array geometry. One array configuration of interest is the *circular array*, whose elements are evenly distributed around a circle. We compare the beampattern of a circular array (blue) and a random array (green) in Fig. 5.6. The radius of the circle, and the radius of the disk are equal. The parameters of each beampattern are shown in Table 5.1.

As seen in Table 5.1, the circular array has a slightly higher directivity, but smaller beamwidth compared with the average of a random planar array with similar geometry. This illustrates that little performance is lost on average if the elements are defined by random locations.



(a) Beampattern of circular array compared with a single instance of a random array, showing similar beampatterns in terms of main beam width and sidelobe height.



(b) Beampattern of circular array compared with the average of 5000 random array beampatterns near the main beam to illustrate the difference in beamwidth shown in Table 5.1

Figure 5.6: Comparison of circular and random array beampatterns with $N = 100$ elements and $\tilde{R} = 15$ wavelengths.

Table 5.1: Circular array and average random array beampattern parameters with $N = 100$ and $\tilde{R} = 15$.

	HPBW [deg]	Directivity [dB]
Average Planar Array	1.98	18.05
Circular Array	1.37	18.46

5.4 Beampattern Distribution

Until now, we have only reviewed the *average* beampattern parameters for random arrays. This section reviews the beampattern's statistical distribution, specifically the CCDF, or the chance that the power of a random beampattern instance exceeds a threshold in a given direction. To motivate the study, we show the beampattern of 100 random arrays in Fig. 5.7, where we wish to know the probability that the beampattern power of an instance lies within some range.

Ochiai et al. found direct computation of the CCDF computationally demanding,

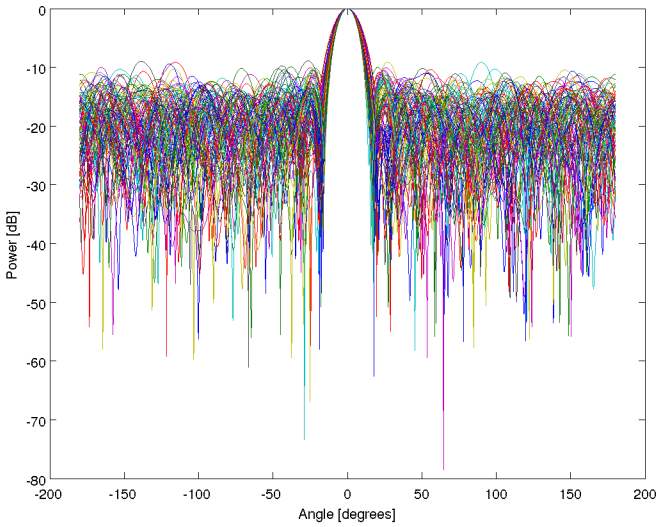


Figure 5.7: Simulated beampattern of 100 random array instances with $N = 64$ elements distributed over $\tilde{R} = 2$ wavelengths.

so they used a guassian approximation for the beampattern distribution, which holds if there a large number of array elements. We show the beampattern CCDF at $\phi_{\text{av}}^{\text{3dB}}$ in Fig. 5.8, which includes the model derived by Ochiai et al. alongside our simulation results. The beampattern instance was more likely to be near to its statistical average as the number of elements increases. The distribution more closely follows the gaussian approximation as the number of transmitters increases. The same model can be applied in the sidelobe region to detect the probability of interference at a certain power level and angle.

We show the beampattern CCDF at all angles in Fig. 5.9 to illustrate the chance that an array instance produces a beampattern power greater than a threshold in all directions. If a designer were concerned with the probability of signal interception, they could use the CCDF to determine the probability that enough power is transmitted for interception to occur.

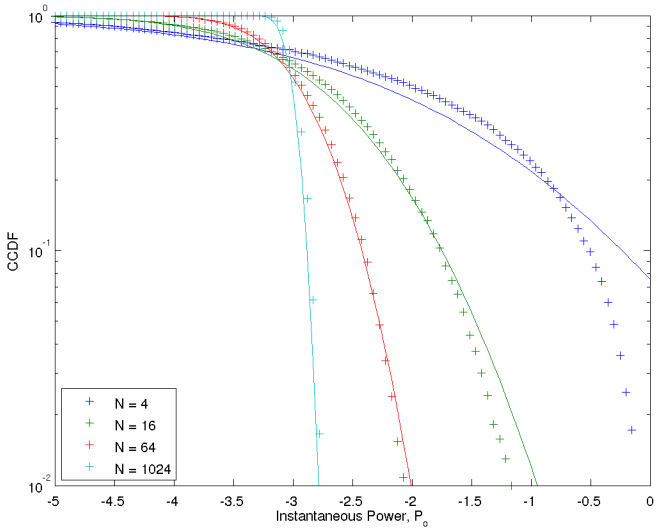


Figure 5.8: Distribution of beampattern when $\phi = \phi_{av}^{3dB}$ for $N = 4, 16, 64, 1024$ elements.

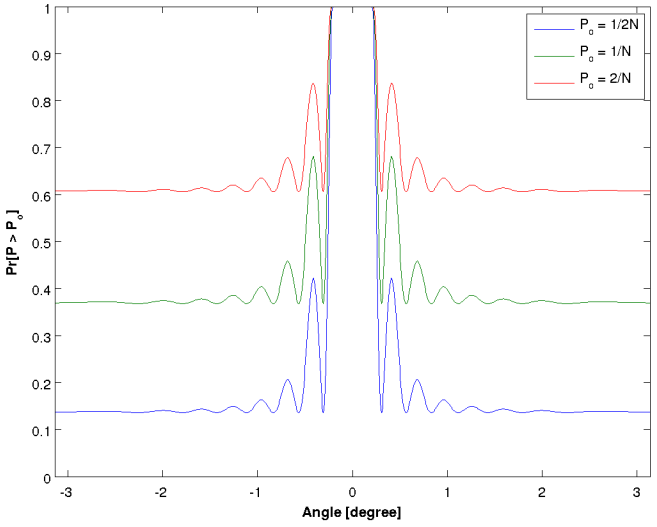


Figure 5.9: CCDF of the instantaneous beampattern power for a threshold power of $P_0 = 1/2N, 1/N, 2/N$ Watts.

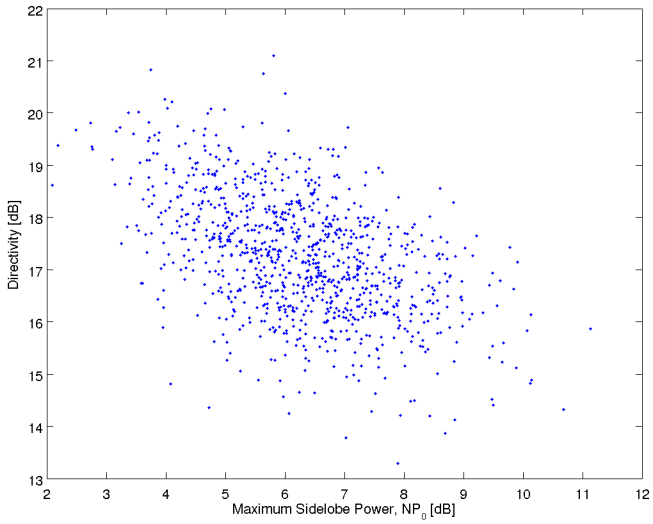


Figure 5.10: Directivity versus the maximum peak sidelobe power NP_0, when $N = 64$ elements and $\tilde{R} = 2$ wavelengths.

5.5 Peak Sidelobes

Occasionally random arrays produce a beampattern containing a large peak away from the main beam. This event decreases the array's directivity, and transmits the signal in unintended directions. Fig. 5.10 shows a weak correlation between sidelobe power and the array's directivity. A 1dB increase in peak sidelobe power corresponds to about a 1dB decrease in directivity.

Ochiai et al. derive a bound for the probability, P_{out}, that the maximum sidelobe exceeds a threshold [17]:

$$P_{out} \leq 4\sqrt{\pi}\tilde{R}\sqrt{NP_0}e^{-NP_0}. \tag{5.11}$$

We verified this bound by simulating 5000 random array instances, and show the results in Fig. 5.11. Our results follow (5.11), but differ for large N. Increasing N decreases the average sidelobe power, responsible for the location of the bounds. This bound would allow a system designer to specify the chance that any sidelobe exceeds

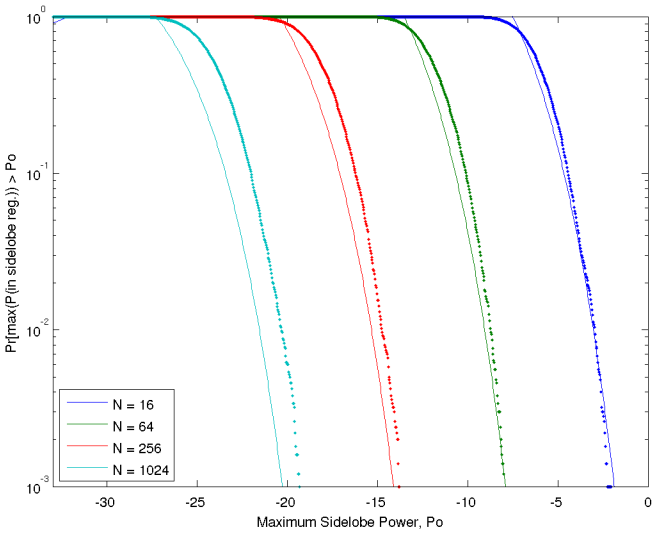


Figure 5.11: Chance of maximum sidelobe power exceeding P_0 using 5000 array instances. The solid line represents the bound in (5.10) and the points show simulation results.

some threshold power. For instance, in our simulations, the maximum sidelobe power exceeded 10 dB of the average sidelobe power only 0.2% of the time.

5.6 Performance of Distributed Beamforming with Imperfect Phase

Until now, we have assumed that the nodes are perfectly syncrhonized in phase and frequency. We now consider the effects of error in the elements of a random array, through their signal phase or location. We study each through simulations, and compare our results with those published by Ochiai et al. We hope to relate our findings within the scope of the overall system design.

5.6.1 Closed-Loop

The synchronization methods described in Chapter 4 control element signal phase without regard for their location. Even the open-loop master-slave method (Sec-

tion 4.2.3) uses a channel estimate to compute its phase delay, so it is considered a closed-loop method in the following analysis. Ochiai et al. reformulate the original array factor by setting the phase in each node to [17]

$$\hat{\Psi}_k = -\frac{2\pi}{\lambda} d_k(\phi_0, \theta_0) + \varphi_k \tag{5.12}$$

where φ_k is an error random variable representing the phase offset in transmitter k. Ochiai et al. found the average far-field beampattern to be [17]

$$P_{av}(\phi) = \frac{1}{N} + \left(1 - \frac{1}{N}\right) \left| 2\frac{J_1(\alpha(\phi))}{\alpha(\phi)} \right|^2 |A_\varphi|^2 \tag{5.13}$$

where

$$A_\varphi \triangleq E_{\psi_k}\{e^{j\varphi_k}\} \tag{5.14}$$

which follows the same form as (5.8), but contains the degradation factor A_ψ, which reduces the beampattern in all directions. Using the phase error distribution, we can determine the average degradation in beampattern power.

Ochiai et al. use the Tikhonov distribution to describe the random variable φ_k, which is commonly used to describe phase jitter in PLL circuits [30]. The Tikhonov distribution is like a circularly distributed gaussian, parametrized by its mean and variance.

We used the Tiknonov distribution to generate random phase errors in the signal phase of each element, then measure the average degradation in mainbeam power. Our results are compared with the degradation factor derived by Ochiai et al. in Fig. 5.12. To maintain an average degradation in mainbeam power less than 3dB, the phase variance must be less than -3dB radians^2.

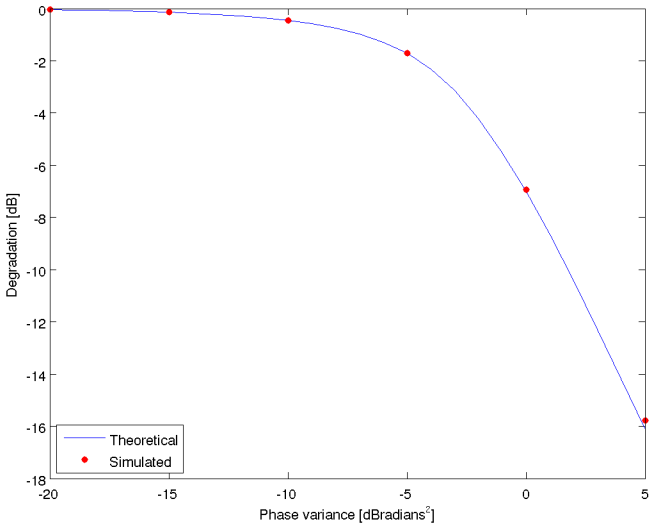


Figure 5.12: Degradation in mainbeam power due to error in element signal phase. Simulation results were measured by taking the average mainbeam power of 5000 realization of a random array with $N = 64$ elements distributed over $\tilde{R} = 2$, with element signal phases perturbed by φ_k.

5.6.2 Open-Loop

True open-loop synchronization would use the element locations for phase synchronization, requiring a level of precision in estimating node locations. Error in node location translates into phase error, reducing the signal coherency at the destination.

Ochiai et al. reformulate the node locations using the polar coordinates $(r_k + \delta r_k, \psi_k + \delta\psi_k)$, where δr_k and $\delta\psi_k$ are error random variables. δr_k was assumed to be uniformly distributed over $[-r_{max}, r_{max}]$ and $\delta\psi_k$ uniformly distributed over $[-\psi_{max}, \psi_{max}]$ [17]. This model has node locations measured from an origin point, perhaps a master node near the center of the configuration.

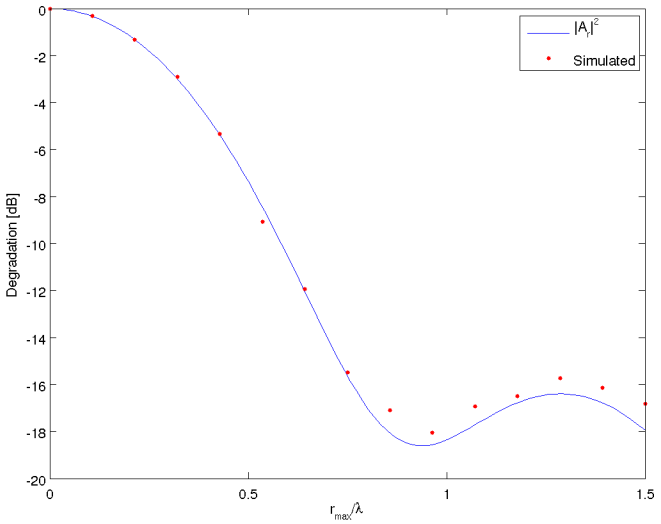


Figure 5.13: The degradation in mainbeam power due to radial location error. Simulation results took average mainbeam power of 5000 random array realization with $N = 64$ elements distributed over an effective area of $\tilde{R} = 2$. The element locations were perturbed by δr_k and $|A_r|^2$ was the degradation factor derived by Ochiai et al.

Ochiai et al. found the average beampattern to be [17]

$$P_{av}(\phi) = \frac{1}{N} + \left(1 - \frac{1}{N}\right) |A_\psi(\phi)|^2 |A_r|^2 \tag{5.15}$$

where A_r and $A_\psi(\phi)$ are degradation factors that limit the beampattern power due to error in the node location's polar coordinates.

A_r, which limits the beampattern due to radial error, was found to be independent of the system parameters N and $\tilde{R}$. Fig. 5.13 shows the degradation factor $|A_r|^2$ derived by Ochiai et al. compared with simulation results. To simulate this scenario, we perturbed the elements' polar locations by δr_k and measured the resulting mainbeam power. To limit the mainbeam degradation below 3 dB, from the figure, we should have $r_{\max} \leq 0.3\ \lambda$. Under this constraint, a 75 MHz system would require a range estimation error less than 1.2 m.

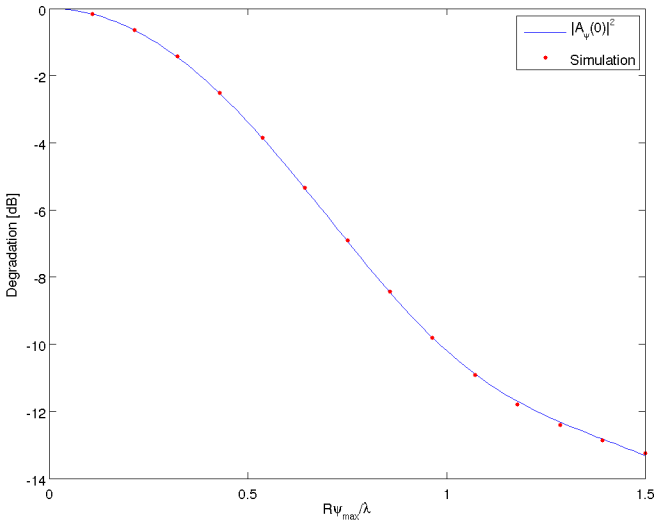


Figure 5.14: Degradation in mainbeam power due to angular location error. Simulation took average mainbeam power of 5000 random array realization with $N = 64$ transmitters distributed over an effective area of $\tilde{R} = 2$. The element locations were perturbed by $\delta\phi_k$ and $|A_\phi(0)|^2$ was the degradation factor derived by Ochiai et al.

The degradation factor $A_\psi(\phi)$ depends on the node distribution spread $\tilde{R}$ and the maximum angular error ψ_{max}. We show the degradation in mainbeam power due to $\psi_{\max}$ in Fig. 5.14, along with simulation results. To limit the mainbeam degradation to below 3 dB, from the figure, we should choose $R\psi_{\max}/\lambda \leq 1/2$. This means the maximum angle estimation error should satisfy [17]

$$\psi_{\max} \leq \frac{\lambda}{2R} = \frac{1}{2\tilde{R}}. \tag{5.16}$$

As $\tilde{R}$ becomes large, the minimum angle uncertainty requirement becomes severe.

The results in this section reveal that a high amount of precision is necessary to synchronize nodes using their location for adequate beamforming gains. As (5.15) indicates, the error due to δr_k and $\delta\psi_k$ both contribute to mainbeam degradation. This means the constraints on $r_{\max}$ and $\psi_{\max}$ hold if the other coordinate estimate

is perfect, making the bound on location ambiguity more stringent, and perhaps impractical.

5.7 Conclusions

In this chapter, we learned that adequate beamforming gains can be achieved with randomly dispersed nodes acting as virtual array elements. By controlling the node distribution, we can control the behavior of their resulting beam. Increasing the nodes' effective spread $\tilde{R} = R/\lambda$ creates a narrower beam with a higher directive gain. However, a narrow beam requires greater accuracy in the the virtual array's pointing angle. Increasing the number of nodes decreases the average sidelobe power, reducing the interference transmitted by the system. However, increasing the node density $N/\tilde{R}$ decreases the average directivity per node, a measure of the system efficiency. The proper tradeoffs must be considered, but the system should be designed to have the narrowest practical beam, with the number of nodes decided by the transmission range and interference limit.

We also studied a model for the beampatter power distribution, which determines the probability that the beampattern power takes a range of values. We studied bounds on the sidelobe power, which could be used to predict the chance of an array instance producing a large peak in an unintended direction. These models could be used to predict the distribution's performance, particularly the probability of signal interception.

We reviewed the effect of element error on beamforming performance and can predict the degradation in beamforming performance due to signal phase error. We also reviewed the possibility of a true open-loop synchronization, which uses node locations to compute the appropriate element signal phase. Our review showed stringent bounds on location estimation error for adequate beamforming performance.

In the next chapter, we design example applications using the information presented in this chapter. A synchronization scheme will be chosen to limit the beamforming degradation. Then, we will configure the node distribution to achieve an average performance that compensates for all errors. Using the simulation framework, we can test our design and simulate its performance.

6. Design and Analysis of a Collaborative Beamforming System

6.1 Introduction

By demonstrating the previously discussed topics through design examples, we hope to provide insight into the dynamics behind collaborative beamforming systems. We design a baseline example, which follows the system model presented in Chpater 5. Then, we alter the model and determine the costs associated with maintaining robust transmission performance.

Collaborative beamforming provides the greatest benefit to wireless sensor networks operating in an open space, with nodes limited in power and the need to transmit information over a large distance. Area monitors, which collect data to monitor a phenomenon and transmit to a base station, could potentially benefit from using collaborative beamforming. Our design examples are inspired by radio controlled vehicles, which move about an area and transmit information to a destination miles away.

We have learned that by controlling the node distribution, the beamforming performance can be modified. Synchronization methods allow the nodes to act as a virtual antenna array. Our examples introduce realistic constraints to review the feasibility of implementing a collaborative beamforming system.

This chapter is organized as follows. Section 6.2 provides a background for our design examples and introduces constraints common throughout this chapter. Section 6.3 outlines the measures used to quantify system performance. Section 6.4 analyzes a baseline design example. Section 6.5 and Section 6.6 introduces alterations to the system model, and their effect on transmission performance. In Section 6.7 we review the examples and discuss their results.

6.2 System Background

Remote vehicles, deployed uniformly over an area and equipped radio transmitters, make up the nodes of a wireless sensor network, forming a virtual antenna array to perform collaborative beamforming.

6.2.1 Nodes

We have a maximum of 100 vehicles free to move about an area, whose locations are modelled by a uniform distribution over a disk. To simplify analysis, the vehicles do not transmit when they are moving, but their random locations introduce randomness to the resulting beamforming performance. Directivity plays a key role in determining the system's transmission performance, and due to its randomness we introduce a margin M_{rand} to ensure a robust link.

We will review the performance of transmitters with carrier frequencies of 75, 900, and 2400 MHz. The nodes have a maximum transmit power of 20 dBm, comparable with portable electronics, and have the ability to control their transmit power. Each transmitter is equipped with a short monopole antenna, which usually operates with a gain on the order of 1 dBi [3]. To simplify analysis, we set the antenna gain to 0 dBi, which could also account for any inefficiencies.

6.2.2 Destination

The destination is defined by its receiver sensitivity S, the minimum signal power required to process an incoming signal. The direction of the destination in relation to the nodes is known, but we account for pointing angle error by introducing the margin $M_{\text{BW}} = 3$ dB, which requires the pointing angle be accurate within the virtual array's 3-dB beamwidth.

To ensure a robust link to the destination, the margins are added to the receiver

sensitivity. A final margin M_{sync} accounts for beamforming degradation due to synchronization errors. This gives a minimum received signal power

$$P_{\text{rx}} = S + M_{\text{rand}} + M_{\text{BW}} + M_{\text{sync}} \tag{6.1}$$

By setting sensitivity $S = -85$ dBm, $M_{\text{rand}} = 5$ dB, and $M_{\text{BW}} = M_{\text{sync}} = 3$ dB, the minimum received signal power is $P_{\text{rx}} = -74$ dBm. We use P_{rx} to determine beamforming system parameters necessary for reliable communications.

6.2.3 Synchronization

In Section 5.6.1, we showed that limiting the phase variance to less than 0.5 radians^2 limits the mainbeam degradation to less than 3 dB. The following system parameters for open-loop master-slave synchronization (Section 4.2.3) satisfy our selection for M_{sync}. The parameters ensure the PLL properly syncrhonizes the nodes' signal

Table 6.1: Open-loop master-slave synchronization system parameters

Parameter	Description	Value
T_1	Sync timeslot duration	0.1 ms
T_2	Transmit timeslot duration	0.9 ms
ω_n	Loop bandwidth	100 kHz
ρ	Phase offset decrease factor	1 %
N_p	Oscillator internal noise spectral density	-101 dBc/Hz

phase, and the maximum phase variance is 0.294 radians^2.

6.3 Performance Measures

The following metrics were used to measure the performance of our design examples.

1. Received Signal Power at Range: We computed the received signal power at the desired range of the system. For an effective transmission link, the received signal power must exceed S at the specified range.

2. Transmission Cost: The per-vehicle transmit power. If the vehicles are constrained by energy capacity, limiting the per-vehicle transmit power maximizes network lifetime. We also compute the nodes' combined transmit power to provide a measure for the system efficiency.

3. Pointing Angle Accuracy: We constrain the pointing angle accuracy to satisfy M_{BW}. We also compute the received signal power at the beamwidth threshold to ensure the constraint is satisfied.

4. Area Interference: We average the received signal power at random locations in the transmit area to obtain a measure for the transmission interference.

Some performance measures are used to constrain the system and motivate design examples. We compare the expected results of these measures with simulations of random instances.

6.4 Example 1 - Baseline Design

Remote operated vehicles must transmit information to a destination 25 km away. The angle to the destination is known within 1°, requiring a 3-dB beamwidth of at

Table 6.2: Distribution radius for each carrier frequency

Carrier Frequency [MHz]	Distribution Radius [m]
75	60
900	5
2400	1.875

least 2°. The distribution area controls the beamwidth by (5.9)

$$\phi_{\text{av}}^{\text{3dB}} = 4 \arcsin\left(\frac{0.1268}{\tilde{R}}\right) > 0.0349 \text{ radians}, \tag{6.2}$$

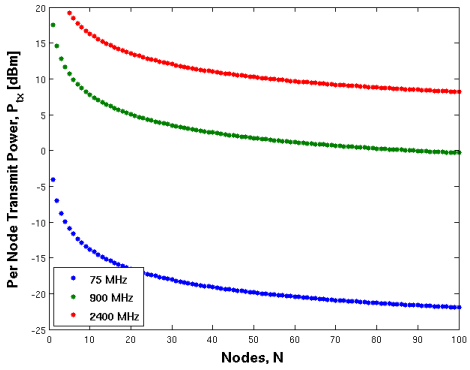
where $\tilde{R} < 14.5\lambda$ provides a sufficiently wide beamwidth. We chose $\tilde{R} = 15\lambda$ to form a good beam and protect against extra error. The distribution area for each carrier wavelength is shown below in Table 6.2

To transmit sufficient signal power, we must choose the number of nodes to deploy and their transmit power. Using the free-space path loss (2.6) and the bound on directivity (5.10), we compute the necessary per-vehicle transmit power for a given number of nodes, namely

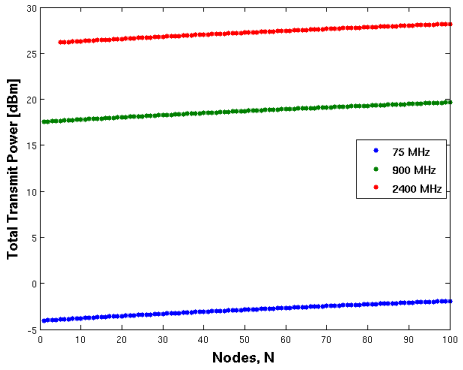
$$P_{\text{tx}} = P_{\text{rx}} \left(\frac{1 + \mu \frac{N}{\tilde{R}}}{N}\right) \left(\frac{4\pi d}{\lambda}\right)^2. \tag{6.3}$$

The tradeoff is illustrated in Fig. 6.1(a). Higher carrier frequencies limit the transmission range of the system and require a greater transmit power to reach the same destination. The system's total transmit power is shown in Fig. 6.1(b). Using fewer nodes is more efficient; the total transmit power varies by 2 dB over the whole constraint set.

We simulated two alternatives for each carrier frequency, one with a minimal number of nodes, and one with the maximum number of nodes (Table 6.3).



(a) Transmission power per-vehicle versus the number of vehicles deployed to achieve adequate signal power at a destination 1 km away.



(b) Total system transmit power. It was found by mulitplying the per-vehicle transmit power on the left by the number of transmitters.

Figure 6.1: Tradeoff between number of nodes deployed and their transmit power.

Table 6.3: Baseline design alternatives

Frequency	Distance	Vehicles	Per Node Transmit Power
75 MHz	25 km	$N = 20$	$P_{\text{tx}} = 11.4$ dBm
		$N = 100$	$P_{\text{tx}} = 6.03$ dBm
900 MHz	5 km	$N = 20$	$P_{\text{tx}} = 19.0$ dBm
		$N = 100$	$P_{\text{tx}} = 13.6$ dBm
2400 MHz	2 km	$N = 20$	$P_{\text{tx}} = 19.6$ dBm
		$N = 100$	$P_{\text{tx}} = 14.2$ dBm

6.4.1 Results

Using the array beampattern and gain, we computed the received signal power in an area covered by the transmission range of the system. Synchronization errors were incorporated by randomly perturbing the nodes' signal phase. For a single random instance of each alternative in Table 6.3, we show the received signal power along the main beam in Fig. 6.2

For each alternative, we simulated the average received signal power and area

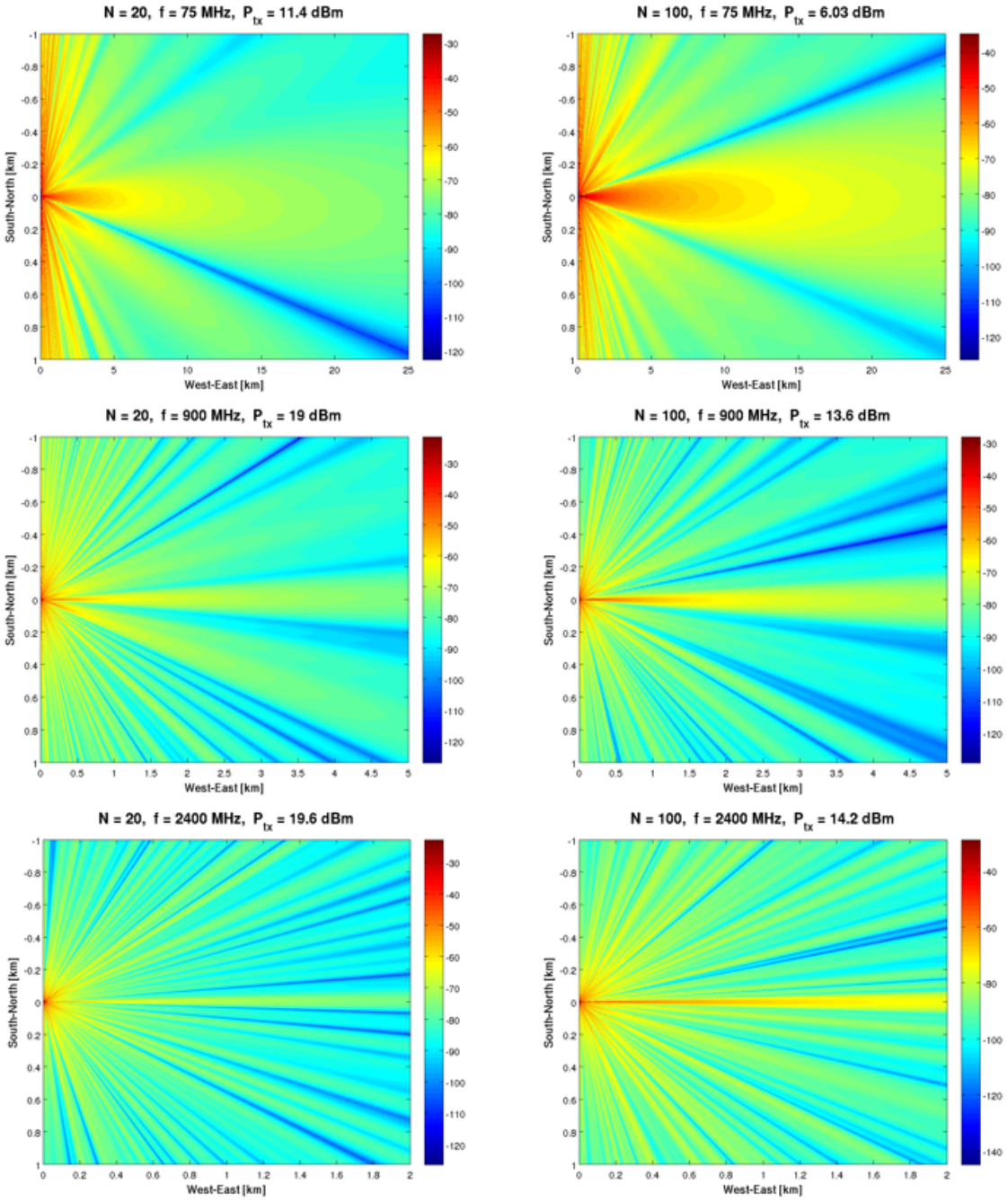


Figure 6.2: Received signal power along main beam. The nodes are located at the left-center of each plot

interference for 1000 random trials. The results are shown in Table 6.4.

Table 6.4: Baseline design simulations: P_{tx} is the per node transmit power, $P_{\text{rx}}^{\text{av}}$ is the average received signal power at the destination, σ is the standard deviation of the received signal power in dB, and I is the average area interference

Frequency	Distance	N	P_{tx} [dBm]	$P_{\text{rx}}^{\text{av}}$ [dBm]	σ [dB]	I [dBm]
75 MHz	25 km	20	11.4	-75.25	0.593	-76.32
		100	6.03	-74.88	0.300	-81.62
900 MHz	5 km	20	19.0	-75.28	0.590	-74.82
		100	13.6	-74.91	0.315	-81.19
2400 MHz	2 km	20	19.6	-75.22	0.583	-76.10
		100	14.2	-74.88	0.301	-81.70

The received signal power at the destination was always enough for effective communications, which verifies our selection for the number of vehicles and their transmit power. The alternatives with 100 nodes produced less average area interference, which was expected because more nodes decrease the sidelobe power level.

6.5 Example 2 - Node Loss

Our next example reviews the effect of lost collaborating nodes, which can occur when the nodes lose power or are unable to transmit to the destination through a clear path. We use the previous example with 100 nodes and a carrier frequency of 900 MHz as a starting point.

To simulate the degradation in transmission performance, we computed the received signal power when the number of collaborating nodes was reduced and the per node transmit power remained constant. The results are shown in Table 6.5. We calculated the expected degradation using the decrease in the number of nodes, and

Table 6.5: Degradation in transmission performance when nodes are lost

Node Loss [percent]	Expected Degradation [dB]	Expected Rx Power [dBm]	Simulated Rx Power [dBm]
0	0	-74.9	-74.9
10	0.45	-75.4	-75.3
25	1.25	-76.1	-75.8
50	3	-77.9	-77.2
75	6	-80.9	-79.8
90	10	-84.9	-83.5

the results agreed with expectations within 1.5 dB. In all cases, the average received signal power was greater than the sensitivity of the destination receiver (-85 dBm). The message will still reach the destination even if 90% of the nodes are unable to transmit.

We would still like to compensate for the 'lost power' to ensure reliable communications. We increase the per-node transmit power using the relation in (6.3).

In Table 6.6, we show the results of compensating for node loss, which were collected by simulating 1000 random array instances and calculating the average received signal power at the destination. Our goal was to keep received power constant irre-

Table 6.6: Results of compensating for node loss

Node Loss [percent]	Compensated Node Tx Power [dBm]	Rx Power [dBm]
0	13.6	-74.9
10	13.9	-74.9
25	14.5	-75.0
50	15.7	-75.1
75	18.2	-75.3
90	21.8	-75.3

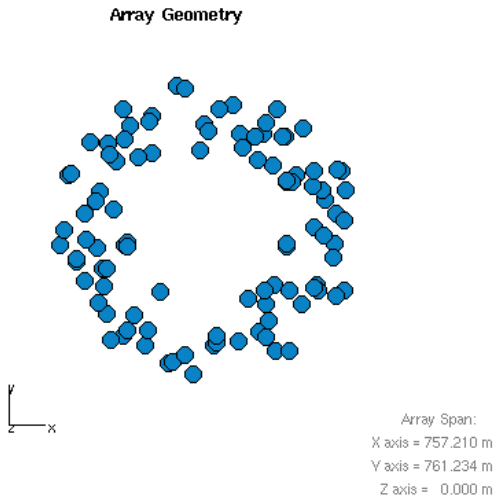


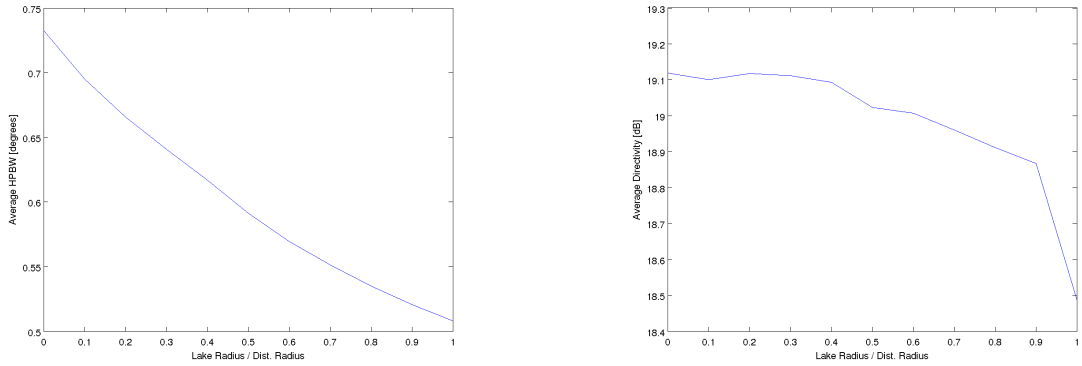
Figure 6.3: Instance of randomly placed nodes around a lake. The radius of the lake is 25% of the distribution radius.

spective of the node loss. The cost of losing nodes is additive node transmit power, which lessens the network lifetime. If 90% of the nodes are lost in this configuration, the compensated per node transmit power is greater than the maximum per node transmit power (we allowed only 20 dBm of transmit power in the problem specification; we require 21.8 dBm).

6.6 Example 3 - Lake Obstruction

We now alter the distribution of nodes by placing an obstruction in the middle of them. The obstruction does not impede a clear path to the destination, so we say the nodes are distributed around a lake. Now, the distribution is uniform on a disk with its center removed. An instance is shown in Fig. 6.3.

Given the new distribution, we want to determine its effect on beamforming performance to ultimately compensate if necessary. We computed the average directivity and 3-dB beamwidth versus lake size, and show the results in Fig. 6.4.



(a) Average 3-dB Beamwidth

(b) Average Directivity

Figure 6.4: Average beam parameters of 100 realizations of a virtual array with randomly placed element on a disk around another disk, which represents a lake.

Directivity does not degrade much as the lake size increases (∼1 dB), and several random instances showed little effect on the received signal power. However, the 3-dB beamwidth decreases by as much as 30%.

We will attempt to compensate for the narrow beamwidth by increasing the node distribution area. Despite the fact that the node distribution has changed, we will still assume that its beamwidth ϕ_i can be computed by [17]

$$\phi_i = 4 \arcsin\left(\frac{0.1268}{\tilde{R}_i}\right) \tag{6.4}$$

where $\tilde{R}_i = R_i/\lambda$. Given an initial beamwidth and spread ϕ_1 and $\tilde{R}_1$, respectively, we can potentially compensate using the relation

$$\frac{\phi_2}{\phi_1} = \frac{\tilde{R}_1}{\tilde{R}_2} \tag{6.5}$$

where ϕ_2 and $\tilde{R}_2$ are the new beamwidth and distribution spread, respectively, given a small argument for $\arcsin(\cdot)$.

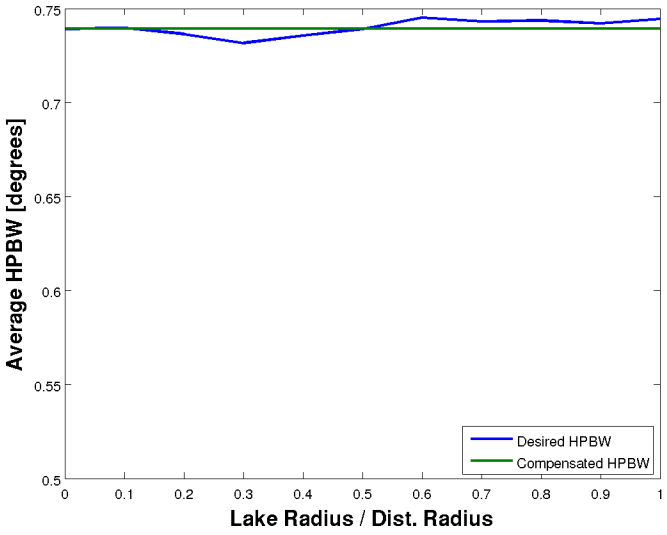


Figure 6.5: Compensated virtual array 3-dB beamwidth with distribution obstructed by a lake

Using the results in Fig. 6.4(a) for ϕ_1, $\tilde{R}_1 = 40$, and the relation in (6.5), we were able to successfully control the new array's beamwidth. The results in Fig. 6.5 show the average 3-dB beamwidth only varies by at most 0.0171°.

It should be noted that decreasing $\tilde{R}_2$ to compensate for a narrowed beamwidth only applies to a point. Since the compensated $\tilde{R}_2$ decreases as the lake size increases, the lake size will eventually exceed the necessary distribution area. In our example, with $R_1 = 160$ m and $\tilde{R}_1 = 40$, the compensated node distribution area equals the lake area when the lake's radius is 75% of the original distribution radius. Nonetheless, bringing the nodes closer to the lake still decreases $\tilde{R}_2$ and widens the virtual array's beamwidth.

6.7 Conclusions

The presented examples outline the design of a system that uses collaborative beamforming to transmit information over a distance. The examples highlight the

primary system dynamics and illustrate a practical backdrop for potential applications. Our examples used remote vehicles, free to move and monitor an area, that collaboratively transmit information to a distant base station.

Using realistic transmission parameters, we designed a system that transmits to a desired range. Our first example outlined the design of a baseline collaborative beamforming system. We illustrated the tradeoff between the number of nodes deployed and their transmit power, and then reviewed the performance of two alternative designs. We also reviewed the effect of losing nodes and determined how the system could compensate; by increasing the per node transmission power. Our final design example altered the node distribution by placing them around a lake. The altered distribution did not affect directivity, but it did affect the beam's 3-dB beamwidth. We provided a way to compensate for a decrease in beamwidth, by shrinking the node distribution and proved its effectiveness in widening the beam.

These examples have helped identify the primary actors in controlling a collaborative beamforming system and a starting point for system design.

7. Conclusions

This thesis reviews collaborative transmit beamforming, a way in which distributed transmitters work together to send a common message. In a wireless sensor network, collaborative beamforming would allow the network to transmit information over greater distance, at a higher rate and more energy efficiently.

We began by reviewing antenna arrays, which can be extended to study virtual arrays with elements detached in space. An application was developed to support research through simulation, which given a model for the array geometry can simulate the beampattern of random virtual arrays and compute parameters associated with beamforming. The simulation framework was verified with results found in the literature.

Synchronization methods allow detached transmitters to work as a virtual antenna array by synchronizing them in phase and frequency. We reviewed researched methods, and the amount of error they introduce to a collaborative beamforming system.

The work of Ochiai et al. [17] was reviewed to gain an understanding for the behavior of virtual arrays whose elements are defined by a random distribution. Using our simulation framework, we verified the findings of Ochiai et al., and concluded that adequate beamforming gains can be achieved using random arrays and including synchronization errors.

Finally, we outlined the design of a system to utilize collaborative beamforming. We modelled the transmission performance of distributed remote-operated vehicles that use collaborative beamforming to transmit information to a destination 25 km away using feasible amount of power. The example was altered to test its robustness and find ways to compensate for a degradation in performance. The examples helped

illustrate collaborative beamforming system dynamics in practice.

Bibliography

[1] Mohammed F A Ahmed, Student Member, Sergiy A Vorobyov, and Senior Member. Collaborative Beamforming for Wireless Sensor Networks with Gaussian Distributed Sensor Nodes. *IEEE Transactions on Wireless Communications*, 8(2):638–643, 2009.

[2] K. Chakrabarty and S. S. Iyengar. *Scalable Infrastructure for Distributed Sensor Networks*. Springer-Verlag, London, 2005.

[3] Kai Chang. *RF and Microwave Wireless Systems*. Wiley Series in Microwave and Optical Engineering. Wiley, New York, 2000.

[4] Mischa Dohler. *Virtual Antenna Arrays*. PhD thesis, University of London, 2003.

[5] Lun Dong, Student Member, Athina P Petropulu, and H Vincent Poor. A Cross-Layer Approach to Collaborative Beamforming for Wireless Ad Hoc Networks. *IEEE Transactions on Signal Processing*, 56(7):2981–2993, 2008.

[6] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-Grained Network Time Synchronization using Reference. *SIGOPS Op. Sys. Rev.*, 36(SI):147–63, 2002.

[7] D. A. S. Fraser. Generalized Hit Probabilities with a Gaussian Target. *Ann. Math. Statis.*, 22(2):248–258, 1951.

[8] John D. Kraus and Ronald J. Marhefka. *Antenna Basics*. Springer-Verlag, 2001.

[9] T Lo. Mathematical Theory of Antenna Arrays with Randomly Spaced Elements. *IEEE Transactions of Antennas and Propogation*, 12(3):257–268, 1961.

[10] Heinrich Meyr, Marc Moeneclaey, and Stefan A Fechtel. *Digital Communication Receivers*. 1998.

[11] R. Mudumbai, G. Barriac, and U. Madhow. On the Feasibility of Distributed Beamforming in Wireless Networks. *IEEE Transactions on Wireless Communications*, 6(5):1754–1763, May 2007.

[12] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac. Scalable feedback control for distributed beamforming in sensor networks. *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 137–141, 2005.

[13] Raghuraman Mudumbai, D. Richard Brown III, Upamanyu Madhow, and H Vincent Poor. Distributed Transmit Beamforming : Challenges and Recent Progress. *IEE Communications Magazine*, (February):102–110, 2009.

[14] Raghuraman Mudumbai, Upamanyu Madhow, Patrick Bidigare, and Rick Brown. DSP-CENTRIC ALGORITHMS FOR DISTRIBUTED TRANSMIT BEAMFORMING. In *Signals, Systems and Computers (ASILOMAR)*, pages 93–98, 2011.

[15] Raghuraman Mudumbai, Ben Wild, Upamanyu Madhow, and Kannan Ramchandran. Distributed Beamforming using 1 Bit Feedback : from Concept to Realization. In *Allerton Conf. Commun., Control, Comp.*, pages 2981–93, Monticello, IL, 2006.

[16] Seo Munkyo, Mark Rodwell, and Upamanyu Madhow. A feedback-based distributed phased array technique and its application to 60-GHz wireless sensor network. In *2008 IEEE MTT-S International Microwave Symposium Digest*, pages 683–686. Ieee, June 2008.

[17] Hideki Ochiai, Patrick Mitran, Student Member, and H Vincent Poor. Collaborative Beamforming for Distributed Wireless Ad Hoc Sensor Networks. *IEEE Transactions on Signal Processing*, 53(11):4110–4124, 2005.

[18] J S Przemieniecki. *Mathematical Methods in Defense Analyses Third Edition*. 3rd edition, 2000.

[19] Francois Quitin, Muhammad Mahboob Ur Rahman, Raghuraman Mudumbai, and Upamanyu Madhow. Distributed beamforming with software-defined radios: Frequency synchronization and digital feedback. *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 4787–4792, December 2012.

[20] David C Rife and Robert R Boorstyn. Single-Tone Parameter Estimation from Discrete-Time Observations. *IEEE Transactions on Information Theory*, IT-20(5):591–8, 1974.

[21] NAVAIR Electronic Warfare/Combat Systems. *Electronic Warfare and Radar Systems Engineering Handbook*. Number April 1997. 2012.

[22] The MathWorks Inc. Phased Array System Toolbox , 2013.

[23] Fung-I Tseng and David K. Cheng. Gain Optimization for Arbitrary Antenna Arrays Subject to Random Fluctuations. *IEEE Transactions of Antennas and Propogation*, 15(3):356–366, 1967.

[24] Ymg-szu Tu and Gregory J Pome. Coherent Cooperative Transmission From Multiple Adjacent Antennas To a Distant Stationary Antenna Through AWGN Channels. In *Vehicular Technology Conference*, pages 130–134, 2002.

[25] Wayes Tushar and David B Smith. Distributed Transmit Beamforming Based on a 3-Bit Feedback System. In *Signal Processing Advances in Wireless Communications (SPAWC), 2010 IEEE Eleventh International Workshop on*, number 1, pages 1–5, 2010.

[26] Wayes Tushar, David B. Smith, Walid Saad, and H. Vincent Poor. Distributed transmit beamforming: Performance improvement using a two-bit feedback scheme. In *2012 International Symposium on Communications and Information Technologies (ISCIT)*, pages 492–496. Ieee, October 2012.

[27] Harry L Van Trees. Arrays and Spatial Filters. In *Optimum Array Processing*, chapter 2, pages 72–89. 2002.

[28] Hubregt J Visser. The Linear Broadside Array Antenna. In *Phased Array Basics*. 2005.

[29] Hubregt J Visser. The Linear Phased Array Antenna. In *Array and Phased Array Antenna Basics*, chapter 7, pages 201–221. 2005.

[30] A. J. Viterbi. *Principles of Coherent Communication*. McGraw Hill, New York, 1966.

[31] Slim Zaidi and Sofiene Affes. Distributed collaborative beamforming with minimum overhead for local scattering environments. In *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1–7. Ieee, August 2012.

[32] Keyvan Zarifi and Ali Ghrayeb. Collaborative Null-Steering Beamforming for. *IEEE Transactions on Signal Processing*, 58(3):1889–1903, 2010.