BEAMFORMING FOR ANTENNAS ON GENERAL WEDGE- AND CYLINDRICAL-SHAPED SURFACES

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Irfanullah

In Partial Fulfillment
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

December 2013

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Beamforming for Antennas on General Wedge- and Cylindrical-Shaped Surfaces

By

Irfanullah

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Benjamim D. Braaten
Chair

Dr. David A. Rogers

Dr. Daniel L. Ewert

Dr. Alan Denton

Approved:

12/12/2013
Date

Dr. Scott Smith
Department Chair

ABSTRACT

Adaptive beamforming antennas capable of accommodating the growing number of wireless subscribers throughout the world has become an essential part of modern wireless systems. In this work, the beamforming of a conformal antenna on a changing conformal surface is studied by relating the varying position of each antenna element in the array to the weighting coefficients (i.e, array weights) required to give a desired antenna beamformation. In particular, the beamforming of a 1×4 array on a changing wedge- and cylindrical-shaped surface is studied using the projection method on a wavefront of a transmitted wave in a particular direction. To validate the theory, a 1×4 prototype antenna with individual voltage-controlled phase shifters and attenuators is used to implement the computed weights of each individual antenna element for measurements in an anechoic chamber. Overall agreement between theory, simulations and measurements is shown throughout the work. Furthermore, the effects of mutual coupling and changing conformal surfaces on the behavior of the beamforming pattern and array weights is investigated and summarized.

ACKNOWLEDGMENTS

I am very grateful to take this opportunity to thank Dr. Benjamin D. Braaten, Dr. David A. Rogers, Dr. Daniel Ewert, and Dr. Alan Denton for serving on my graduate committee. I cannot thank Dr. Braaten enough for the guidance, encouragement and wisdom he gave me during my graduate studies. He has shown me the joys of teaching, rewards of research and importance of life, and for this I thank him. I would especially like to thank Dr. David A. Rogers who has been an invaluable part of my research and teaching during my graduate studies at North Dakota State University. I would like to thank Dr. Daniel Ewert and Dr. Alan Denton for serving on my graduate committee. I appreciate the comments and help you have provided as well as the willingness to serve on my committee with such short notice. Finally, I would like to thank my family for their support and understanding that I had to leave my hometown to pursue this work.

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION AND PREVIOUS WORK

1.1. Background

Usually the radiation pattern of a single radiating element is relatively wide, and each element provides low values of directivity (gain) [1]-[2]. In many applications it is necessary to design antennas with very directive characteristics (very high gains) to meet the demands of long distance communication. Several antennas can be arranged in space and interconnected to produce a directional radiation pattern. Such a configuration of multiple radiating elements is referred to as an array antenna, or simply, an array [3]. Arrays offer the unique capability of electronic scanning of the main beam. By changing the phase of the exciting currents in each element antenna of the array, the radiation pattern can be scanned through space. The array is then called a phased array [4]. Phased arrays have many applications, particulary in radar. The concept of phased arrays was proposed in 1889, but the first successful array (a two-element receiving array) did not appear until about 1906 [5]. The introduction of shortwave radio equipment in the 1920s made possible the use of reasonably sized antenna arrays, providing a convenient way to achieve a directive radiation pattern for radio communications. Around the time of World War II, array antennas operating at VHF, UHF, and later, microwave frequencies were introduced for use in radar systems. Today, arrays are used extensively.

The following section defines and briefly discusses types of antenna arrays, and particularly the concept of a phased array and adaptive antenna arrays. Discussing these topologies here will lay the foundation for the conducted research. The types of array topologies and an adaptive antenna system is described. After this, the idea of smart antennas is presented and illustrated with a functional block diagram.

1.1.1. Geometrical configurations of antenna arrays

Arrays are found in many geometrical configurations. The most elementary is

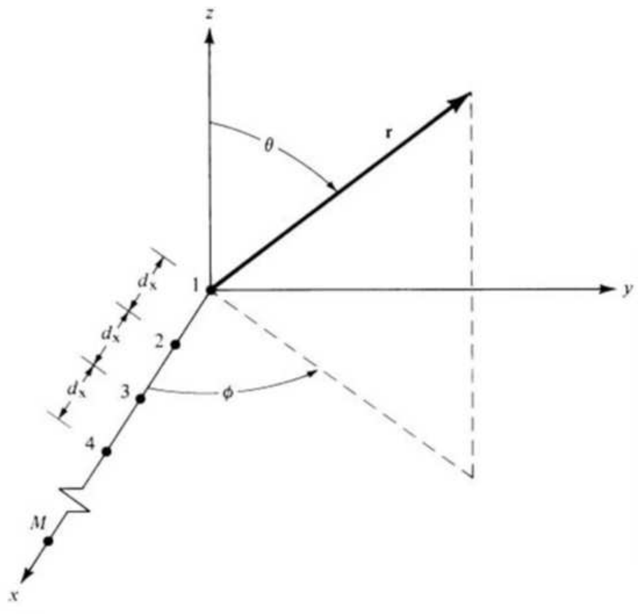

Figure 1. Linear array.

a linear array in which the array element centers lie along a straight line, as shown in Figure 1. The elements in an array can form a planar array. A popular planar array is the rectangular array shown in Figure 2. Furthermore, conformal arrays are yet another class, where the array element locations conform to a nonplanar surface. This is a great advantage for arrays on the skin of a vehicle such as an airplane. Figs. 3 and 4 shows examples of conformal topologies.

An array has many advantages over a single element. For example, the narrow main beam of a parabolic reflector antenna is scanned by slewing the entire structure, whereas arrays can be phase-scanned at the speed of the control electronics without moving the antenna. In addition, it is possible to track multiple targets with a phased array. Weighting the signals before combining them enables enhanced performance features such as interference and beam steering without physically moving the aperture. It is even possible to create an antenna array that can adapt its performance to suit its environment. The price paid for these attractive features

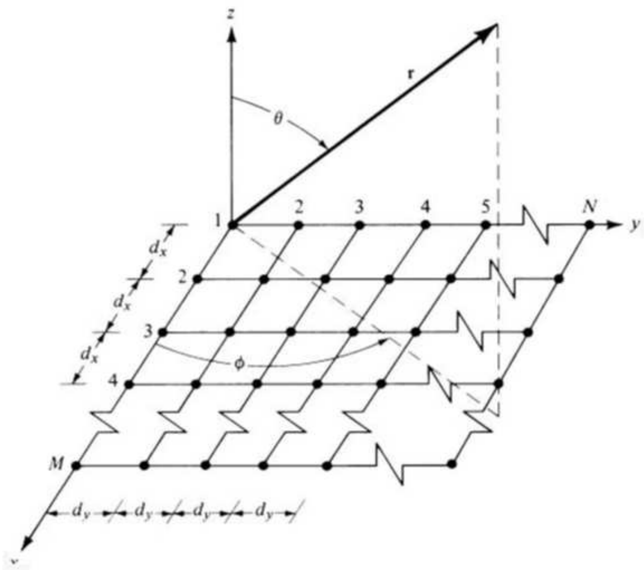

Figure 2. Planar array.

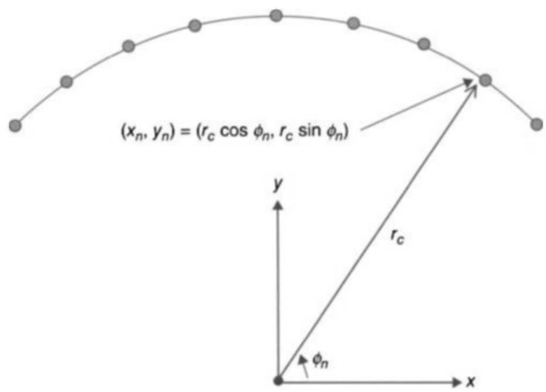

Figure 3. Linear array bent to conform to a circular arc.

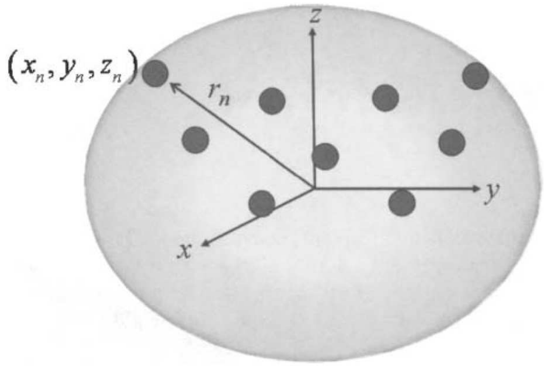

Figure 4. Nonplanar array.

is increased complexity and cost.

For many years, conformal antennas have been used extensively for wireless applications that require an antenna to operate on a surface that is not flat (i.e., a singly or doubly curved surface). Wearable antennas are good examples of conformal antennas that are required to operate over a wide range of surface deformations. Another typical application could be as a base station antenna in a mobile communication system. Today, the common solution is three separate antennas each covering a 120-degree sector. Instead, one cylindrical array could be used, resulting in a much more compact installation and less cost [6].

Much attention has been paid to the design and implementation of adaptive antenna arrays (also called smart antennas), intended for use in current and future wireless communication systems. The motivation is that employment of such array

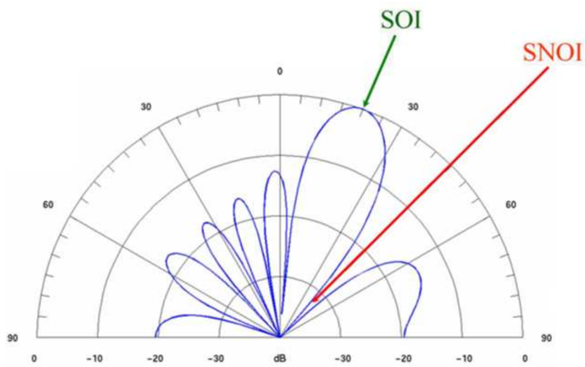

Figure 5. SOI and SNOI in smart antenna system.

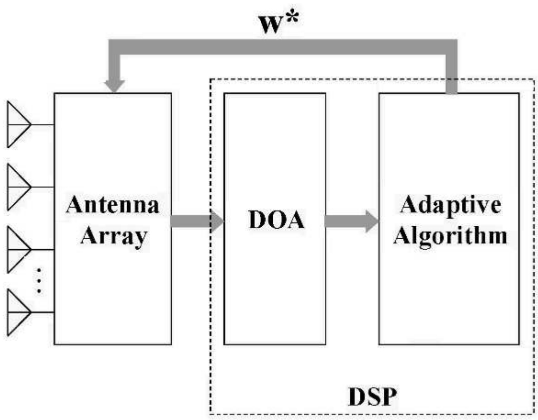

Figure 6. Basic smart antenna system.

antennas can significantly increase the range and capacity of wireless systems. The main feature of smart antennas, compared with fixed-beam antennas are their abilities to form a steered beam and to track mobile users in a sector; which in many applications, has a 120 degree azimuth angular size [7].

As shown in Figs. 5 and 6 and illustrated in [8]-[10], an adaptive antenna system can tune out unwanted interferers by placing nulls toward the signal not of interest (SNOI), and concentrate on the desired user by placing the main beam toward the signal of interest (SOI). The digital signal processor (DSP) computes the direction of arrival (DOA) of the user from the time delay. The DSP also adds the strength of the signals from each antenna element together and forms a beam toward the direction as computed by the DOA.

Smart antennas integrate radio intelligence (DSP) with antenna array technology to (a) enhance communication system performance, including capacity and range and (b) improve link quality, for transmission and reception, by multipath management and mitigation of fading. These are accomplished by beam steering (placing beam maxima toward SOI) and null steering (placing beam minima, ideally nulls, toward SNOI).

1.1.2. Previous work on conformal array antennas

The field of phased array antennas was a very active area of research from WW II up to about 1975 [11]. During this period, much pioneering work was done also for conformal arrays [12]. However, electronically scanned, phased array antennas did not find widespead use until the necessary means for feeding and steering the array became available. Integrated circuit (IC) technology, including monolithic microwave integrated circuits (MMIC), filled this gap, providing reliable solutions with a potential for low cost, even for very complex array antennas. An important factor was also the development of digital processors that can handle the enormously increased rate of information provided by phased array systems. Digital processing techniques made phased array antenna systems cost effective. This being true for phased arrays in general, it also holds for conformal array antennas. However, in the area of conformal arrays, electromagnetic models and design know-how needed extra development. During the last 10 to 20 years, numerical techniques, electromagnetic analysis methods, and the understanding of antennas on curved surfaces have improved [13]-[19].

The origin of conformal arrays can be traced at least back to the 1930s when a system of dipole elements arranged on a circle, thus forming a circular array, was analyzed by Chireix [20]. Later, in the 1950s, several publications on the subject were presented; see for example [21]-[22]. The circular array was attractive because of its rotational symmetry and proper phasing can create a directional beam which can be scanned 360 degrees. The applications were in broadcasting communication and, later, navigation and direction finding. An advanced, more recent application using a large circular array, is the French RIAS experimental radar system [23]-[24].

A great deal of important conformal work was done at the U.S. Naval Electronics Laboratory Center (NELC) in San Diego. The work included development of both cylindrical and conical arrays as well as feeding systems [11]-[12]. An indication of a recent resurgence in the interest in conformal antennas is the series of conformal antenna workshops, held in Europe every second year, starting in 1999. The first was held in Karlsruhe (Germany), the second in the Hague (Netherlands), the third in Bonn (Germany), the fourth in Stockholm (Sweden), the fifth in Bristol (UK) and the seventh in Zagreb (Croatia) in 2011 [25]-[30].

More recently in [31]-[32], the effect of the radius of curvature on the radiation patterns of conformal antennas was presented. Then in 2012, Braaten [33] investigated the radiation pattern of a 1 x 4 microstrip antenna array attached to various conformal surfaces. It was shown using measurements and analytical results that with appropriate phase compensation using the projection method [34], the radiation pattern could be recovered. The projection method presented in [33] for radiation pattern recovery on a conformal surface (wedge) is shown in Figure 7. The linear array is bent from the flat position (original reference plane) and placed on the singly curved surface shaped as a wedge with an angle θ_w. The position of each element on the wedge is represented as a solid black dot with the outline of the surface illustrated as a black line. For this case, if each antenna element is excited with voltages that have the same phase, the E-field radiated from each element will have the same phase. However, when the fields from the elements $A_{\pm 2}$ arrive at the new reference plane, the phase will be different than the fields radiated from elements $A_{\pm 1}$. This phase difference is due to the free-space propagation in the y-direction. Because these phases are not the same, the radiation may not necessarily be broadside to the array (i.e., in the z-direction with $\phi_s = \pi/2$). The required phase compensation using the projection method can be computed as

$$\Delta\phi_n^w = +kL|n|\sin\theta_b \tag{1.1}$$

where $\theta_b = (\pi - \theta_w)/2$ is the bend angle of the array. The uncorrected and corrected radiation pattern using the projection method for the conformal array in Figure 7 is shown in Figure 8.

This work was then extended further in [35]-[36] to study the effects on the radiation pattern of singly curved conformal arrays using amplitude tapering and phase tapering independently. A vision of a future smart skin conformal antenna is shown in Figure 9. This antenna constitutes a complete RF system, including not only the radiating elements but also feed networks, amplifiers, control electronics, power distribution, cooling, filters, and so on, all in a multilayer design that can be

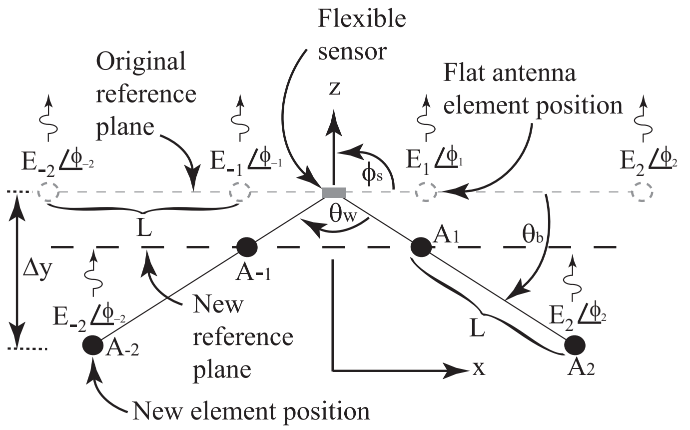

Figure 7. Phase compensation of a linear array on a single curved surface shaped as a wedge.

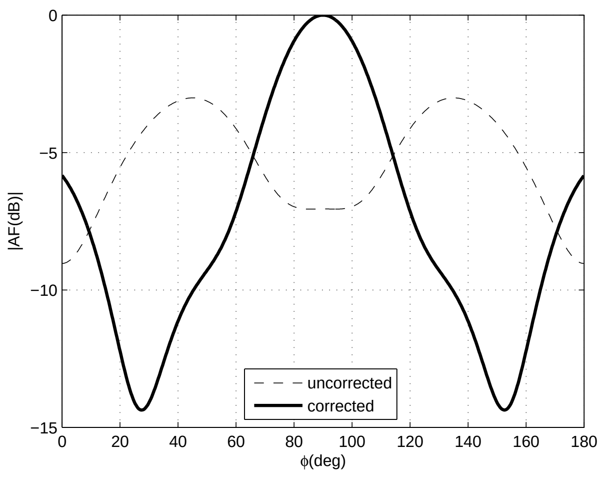

Figure 8. Analytical pattern for the 1 x 4 linear array on a wedge with $\theta_b = 45°$.

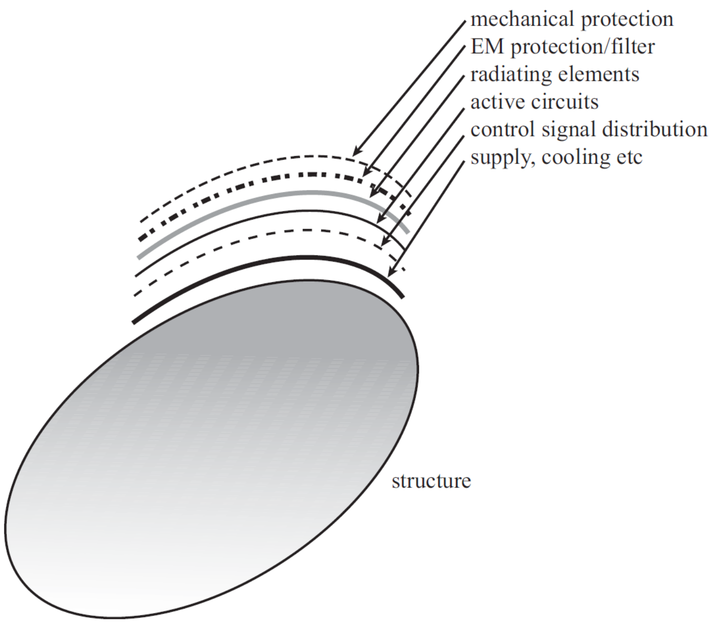

Figure 9. Vision of a smart-skin antenna.

tailored to various structural shapes [6].

The work mentioned above is just a summary of work over the past 40 years that has focused on the radiation patterns of conformal array antennas applied to various shapes (wedge, circular, cylindrical etc). Although the summary presented is not about the detailed idea of amplitude and phase tapering correction, the summary is very important because the foundations for studying the radiation pattern recovery on conformal surfaces is found throughout the work. Many other areas that involve different conformal structures, for example, conical, spherical, feed techniques [11], and microstrip patch configurations [46]-[50] exist and are not discussed here.

1.1.3. Previous work on smart antennas

Over the last decade, wireless technology has grown at a formidable rate, thereby creating new and improved services at lower costs. This has resulted in an increase in airtime usage and in the number of subscribers. The most practical solution to this problem is to use spatial processing. As Andrew Viterbi, founder of Qualcomm Inc., clearly stated: 'Spatial processing remains as the most promising, if not the last frontier, in the evolution of multiple access systems' [60].

Spatial processing is the central idea of adaptive antennas or smart-antenna systems. Although it might seem that adaptive antennas have been recently discovered, they date back to World War II with the conventional Bartlett beamformer [61]. It is only due to today's advancement in powerful low-cost digital signal processors, general purpose processors (and ASICs-Application-Specific Integrated Circuits), as well as innovative software-based signal-processing techniques (algorithms), that smart antenna systems have received enormous interest worldwide. In fact, many overviews and tutorials [60]-[62] have emerged, and a great deal of research is being done on the adaptive and direction-of-arrival (DOA) algorithms for smart-antenna systems [8]-[10] and [44]-[45]. As the number of users and the demand for wireless services increases at an exponential rate, the need for wider coverage area and higher transmission quality rises. Smart-antenna systems provide a solution to this problem.

In adaptive array antenna systems (also called smart antenna systems), special signal processing algorithms are applied to the digitized received signals [55]-[56]. By deriving optimum weights for the element channels, based on the received signal characteristics, the signal-to-noise-plus-interference ratio (SNIR) can be maximized. The result can be seen as creating nulls in the reception pattern in the directions of the interfering sources while preserving a high-gain beam toward the useful signal.

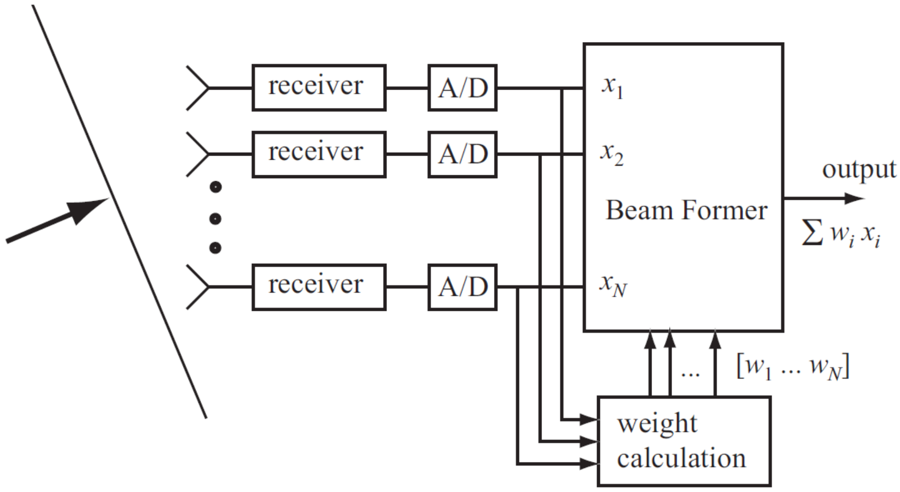

Figure 10. Functional block diagram of an adaptive array system.

A simple schematic for an adaptive beam forming system is shown in Figure 10 [9]. This figure shows that after the system downconverts the received signals to baseband and digitizes them, it locates the SOI using the DOA algorithm, and it continuously tracks the SOI and SNOIs by dynamically changing the weights (amplitudes and phases of the signals). Basically, the DOA computes the direction of arrival of all signals by computing the time delays between the antenna elements, and afterward the adaptive algorithm, using a cost function, computes the appropriate weights that result in an optimum radiation pattern. This is only necessary for DOA-based adaptive beamforming algorithms [43]. However, for reference (or training) based adaptive beamforming algorithms, it does not need the DOA information but instead uses the reference signal, or training sequence, to adjust the magnitudes and phases of each weight to match the time delays created by the impinging signals into the array. In essence, this requires solving a linear system of normal equations [43] and [56]. This linear algebra approach to compute the complex weights for linear beamforming will be explained in Chapter 2. It should be noted that all the work mentioned above was done using the planar antenna arrays for adaptive smart antenna systems. At this point the work by [8]-[10] and [33]-[36] is the closest to the work presented here. This seems to be the first time adaptive antenna arrays (smart antennas) on conformal geometries are considered.

In this section it was shown that significant research is still being conducted on smart antenna arrays. But throughout this summary it was also shown that the case of adaptive beam forming on conformal surfaces has not been studied.

CHAPTER 2. CONDUCTED RESEARCH

There are many properties of interest of conformal and smart antennas. These include array topologies, array factor, resonant frequency, bandwidth, gain, efficiency, far-field patterns, spacing between elements and different feed techniques. From the summary in chapter 1, it can be seen that more recent research involves adaptive beamforming for linear and planar arrays. But throughout all this research many fundamental questions have not been answered about adaptive beamforming for conformal surfaces that change shape with time.

To introduce the conducted research, the two-element adaptive linear antenna array in Figure 11 will be considered. The array factor (AF) expression for an array with N elements, like the one in Figure 4, Chapter 1 with elements distributed in three-dimensional space is given by [55]:

$$AF = \sum_{n=1}^{N} w_n e^{jk[x_n u + y_n v + z_n \cos\theta]}. \tag{2.1}$$

Equation (2.1) assumes a spherical coordinate system where $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$, and w_n are the complex excitation coefficients. Each complex weighting function is defined as $w_n = A_n e^{j\alpha_n}$ where α_n is the phase and A_n is the amplitude of the nth element. The weights for the two element adaptive array shown in Figure 11 can be computed using a linear algebra approach with two equations and two unknowns. The array factor obtained on the basis of these weights is shown in Figure 12. These complex weights were calculated to receive a desired signal (SOI) at $\theta = 0$ (broadside) while simultaneously rejecting the interference signal (SNOI) at angle $\theta = \pi/6$ radians from broadside. The matrix method for linear adaptive beamforming (discussed in next section) along with projection method for conformal antennas (discussed in Section 1.1.2 of chapter 1, and detailed work in [33]) will be

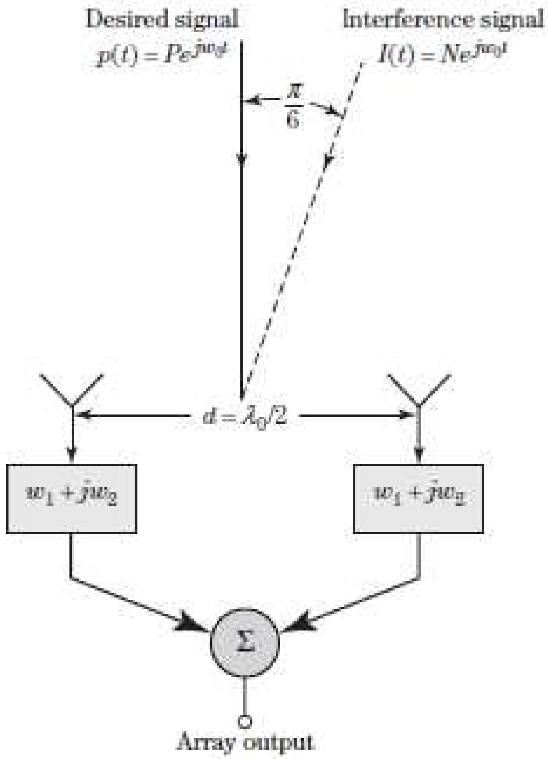

Figure 11. The two-element adaptive array.

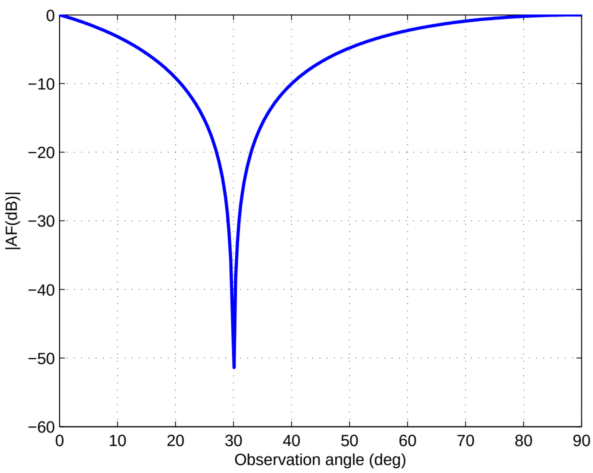

Figure 12. Array factor of the two-element adaptive array in Figure 11.

used for adaptive beamformation on conformal surfaces that change shape with time.

2.1. Linear adaptive beamforming

In this section, the matrix method in [71] and the projection method in [33] and [34] for linear adaptive beamforming is discussed. The array factor of a linear array of N (even) identical elements with uniform spacing positioned symmetrically along the x-axis, as shown in Figure 13, can be written as [71]:

$$\begin{aligned}(AF)_N = {} & w_1 e^{+j(1/2)\psi_1} + w_2 e^{+j(3/2)\psi_2} + ... + w_{N/2} e^{+j[(N-1)/2]\psi_{N/2}} \\ & + w_{-1} e^{-j(1/2)\psi_1} + w_{-2} e^{-j(3/2)\psi_2} + ... + w_{-N/2} e^{-j[(N-1)/2]\psi_{N/2}}.\end{aligned} \tag{2.2}$$

Simplifying and normalizing, (2.2) reduces to

$$(AF)_N = \sum_{n=1}^{N/2} w_n \cos[(2n-1)\psi_n] \tag{2.3}$$

where

$$\psi_n = \frac{\pi d}{\lambda} \sin\theta. \tag{2.4}$$

The angle θ represents the angle of the signal of interest (θ_{SOI}) or the angle of the signal not of interest (θ_{SNOI}) and w_n are the complext excitation coefficients (i.e., array weights) to be determined to give the desired SOI and SNOI radiation pattern for linear antenna array.

To compute the array weights in (2.3), consider the four-element linear array shown in Figure 14. The matrix method for linear adaptive beamforming in [71] is adopted. Assuming one SOI and three SNOIs, four equations with four unknown complex weights can be determined as follows: The output y(t) of the array in Figure 14 due to the desired signal at θ_{SOI} is analyzed first, followed by the output due to

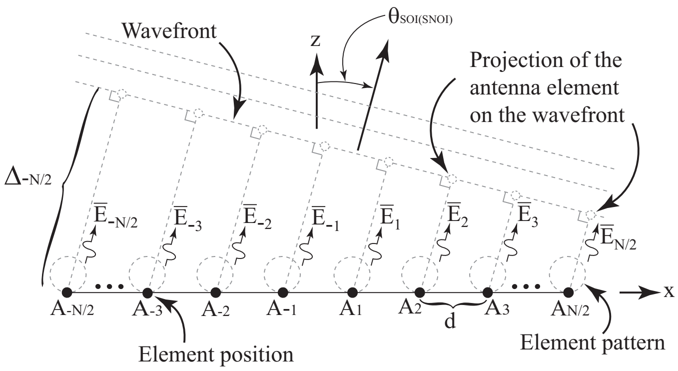

Figure 13. N-element linear array.

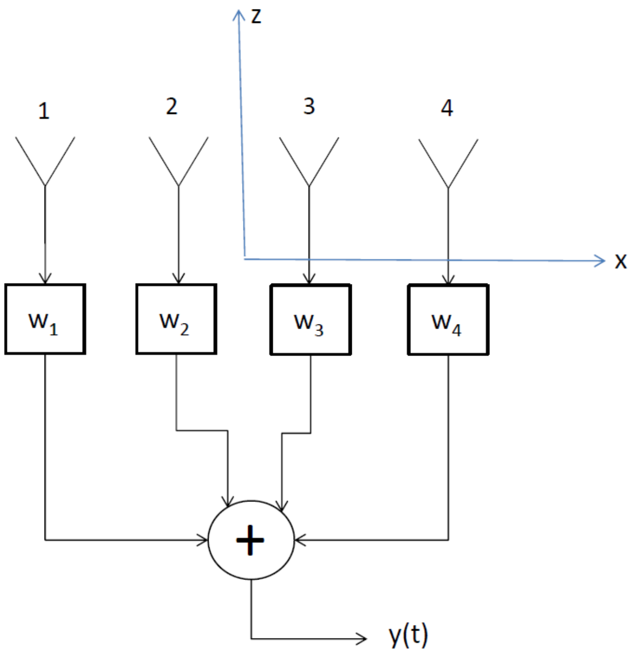

Figure 14. Four element linear array combiner.

the interferers at angles θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3} respectively. Thus, the output y(t) of the array due to the desired signal at θ_{SOI} using (2.2) can be written as

$$y(\theta_{SOI}) = w_1 e^{\pm j(3/2)\psi_1} + w_2 e^{\pm j(1/2)\psi_2} + w_3 e^{\pm j(1/2)\psi_1} + w_4 e^{\pm j(3/2)\psi_2}. \tag{2.5}$$

where the plus or minus sign is associated with phase lead or lag respectively. For the output $y(\theta_{SOI})$ to be equal (unity) only to the desired signal, it is necessary that

$$y(\theta_{SOI}) = 1. \tag{2.6}$$

On the other hand, the outputs y(t) due to the interfering signals are given as

$$y(\theta_{SNOI_1}) = w_1 e^{\pm j(3/2)\psi_1} + w_2 e^{\pm j(1/2)\psi_2} + w_3 e^{\pm j(1/2)\psi_1} + w_4 e^{\pm j(3/2)\psi_2}, \tag{2.7}$$

$$y(\theta_{SNOI_2}) = w_1 e^{\pm j(3/2)\psi_1} + w_2 e^{\pm j(1/2)\psi_2} + w_3 e^{\pm j(1/2)\psi_1} + w_4 e^{\pm j(3/2)\psi_2}, \tag{2.8}$$

and

$$y(\theta_{SNOI_3}) = w_1 e^{\pm j(3/2)\psi_1} + w_2 e^{\pm j(1/2)\psi_2} + w_3 e^{\pm j(1/2)\psi_1} + w_4 e^{\pm j(3/2)\psi_2} \tag{2.9}$$

where ψ_n in (2.5)-(2.9) at each value, θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} or θ_{SNOI_3}, can be evaluated using (2.4). For the output response y(t) to be zero (i.e., reject totally the interference), it is necessary that

$$y(\theta_{SNOI_1}) = (\theta_{SNOI_2}) = (\theta_{SNOI_3}) = 0. \tag{2.10}$$

Solving simultaneously the linear system of four complex equations (2.5), and (2.7)-(2.9), the array weights can be computed. These are the optimum weights that

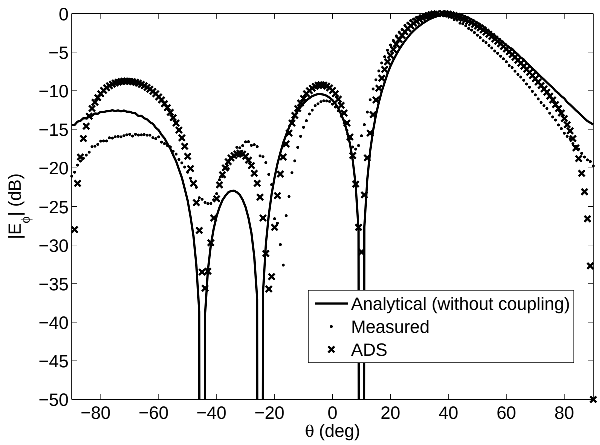

Figure 15. Beamforming results for the four-element linear array.

guarantee the desired SOI and SNOI radiation pattern. The previous expressions were determined for $N = 4$ and can be generalized for more elements. It should be mentioned that similar work could be carried out for an odd value of N.

Next, the beamforming results for the four-element linear array are shown in Figure 15 for $\theta_{SOI} = 40°$, $\theta_{SNOI_1} = -45°$, $\theta_{SNOI_2} = -25°$ and $\theta_{SNOI_3} = 10°$. Good agreement between measurements, simulations and analytical computations is shown. Details of the measurement setup will be discussed in Chapter 7.

In the area of conformal antennas, researchers have studied how the radiation pattern of a conformal antenna changes as it is deformed in various ways [81]-[55]. It has been shown that the overall gain of an antenna array can be reduced by as much as 25 dBi in a particular direction without appropriate phase and amplitude compensation [55]. To mitigate some of the unwanted affects of deformation on the overall gain of an array, several compensation techniques have been proposed [85]-[91]. In summary, it has been shown that the radiation pattern of a conformal antenna attached to a changing surface (i.e., vibrating surface) can be improved with different calibration techniques, signal processing algorithms, sensor circuitry and phase- and amplitude-adjustments. Then again, the previous work was developed under the

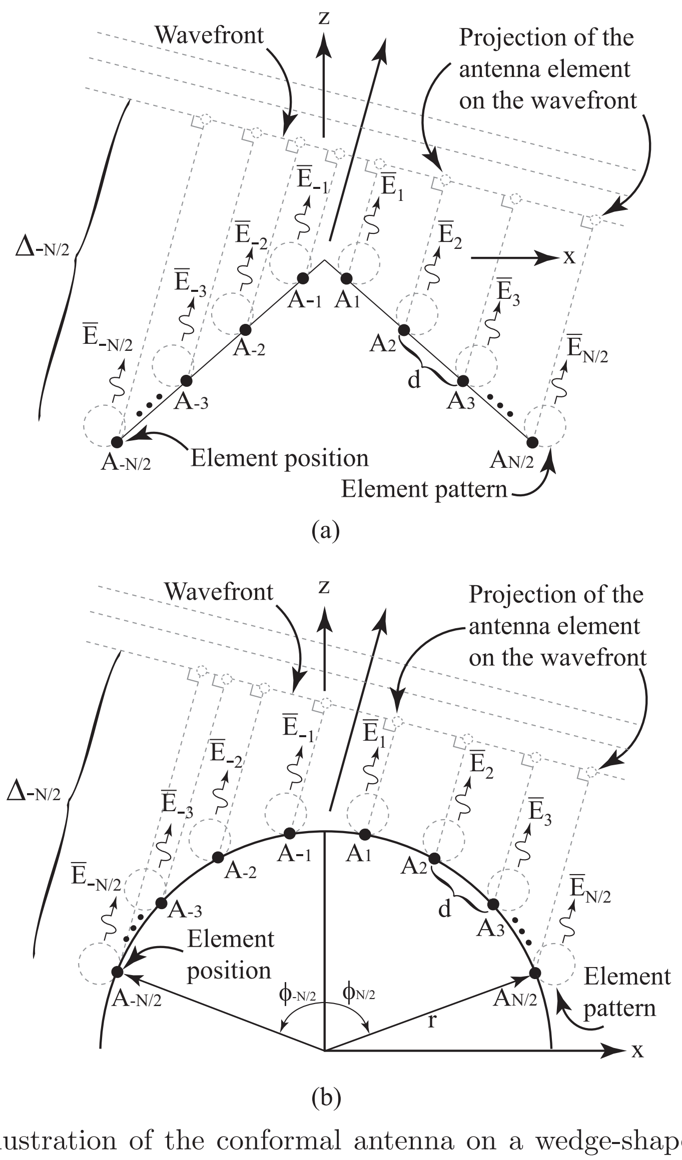

Figure 16. (a) Illustration of the conformal antenna on a wedge-shaped surface and (b) illustration of the conformal antenna on a cylindrical-shaped surface.

assumption that the main lobe direction of the radiation pattern was fixed, and the location of the nulls for beamforming was not defined.

Thus, the objective of this work is to study the theory of beamforming for conformal antennas attached to changing surfaces. Moreover, the changing surface affects on the radiation properties of conformal beamforming antennas will be explored. In particular, this work will focus on beamforming for the one dimensional 1×N arrays on the wedge- and cylindrical-shaped surfaces shown in Figures 16(a) and 16(b), respectively. For both Figures 16(a) and 16(b), the location of each antenna element on the conformal surface is shown as a black dot and denoted as $A_{\pm n}$ where N is assumed to be even and $n = -N/2, ..., -1, 1, ..., N/2$. It should be noted that similar work could be carried out for an odd value of N. Also, the wavefront is denoted as a grey dotted line and the direction of propagation is shown with a black arrow. The wavefront is shown away from the origin for illustrative purposes only.

By developing beamforming for the arrays in Figures 16(a) and 16(b), new applications for smart antennas are possible. One new area is wearable networks and, in particular, wearable antennas [92]. Wearable antennas can be attached to search and rescue personnel, military gear and clothing, and, in many of these applications, the wearable antenna can be deformed due to movement. This can make it difficult to implement a wearable smart antenna for successful wireless communications. By understanding the effects of surface deformation and combining this information with the self-adapting network developed in [85], a wearable smart antenna could be developed. Furthermore, the results presented in this paper could be used to improve the performance of existing smart antennas attached to buildings, vehicles or aircraft that are moving or vibrating.

In the conducted research, the beamforming of a conformal antenna on a changing conformal surface is studied by relating the varying position of each antenna element in the array to the weighting coefficients (i.e, array weights) required to give a desired antenna beamformation. In particular, beamforming for a 1×4 array on a changing wedge- and cylindrical-shaped surface is studied using the projection method on a wavefront of a transmitted wave in a particular direction. To validate the theory, a 1×4 prototype antenna with individual voltage-controlled phase shifters and attenuators is used to implement the computed weights of each individual antenna element for measurements in an anechoic chamber. Overall agreement between theory, simulations and measurements is shown throughout the conducted research. Furthermore, the effects of mutual coupling and changing conformal surfaces on the behavior of the beamforming pattern and array weights is studied, and validated through analytical, full-wave simulations and measurements.

CHAPTER 3. BEAMFORMING OF A 1×4 ARRAY ON A WEDGE-SHAPED CONFORMAL SURFACE

In this chapter the 1×4 array (i.e., $N = 4$) on the wedge-shaped surface shown in Figure 17 with a bend angle of θ_b will be considered first. The inter-element spacing along the surface of the wedge is denoted as d and the location of element $A_{\pm n}$ is denoted as $(x_{\pm n}, z_{\pm n})$.

The motivation for using a beamforming antenna is the feature of being able to define a main radiation lobe in a particular direction while simultaneously defining nulls in the radiation pattern in other directions. The main lobe can then be used to communicate with a desired user in a particular direction and the nulls can be used to reduce the incoming signal from users not of interest. For this work, the angle of the signal of interest will be denoted as θ_{SOI} and one user of interest will be assumed while the angle of the n^{th} user not of interest will be denoted as θ_{SNOI_n}. Because a four-element array is being considered here, three nulls at θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3} are defined. Now, the goal is to define appropriate array weights that can be used to drive the four elements in Figure 17 to give a main lobe at θ_{SOI} and three nulls at θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}. Furthermore, to determine how the shape of the wedge changes the weights, they (weights) are to be determined for various bend angles θ_b of the wedge.

To compute the surface-dependent array weights, the projection method [34] will be used along with the matrix method for computing antenna weights as defined in [71]. Since a maximum radiation in the direction of θ_{SOI} and nulls in the directions of θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3} are desired, the array will be considered as a transmitter in Figure 17. First, consider the case when the array is radiating in the direction of θ_{SOI}, as shown in Figure 17(a). One method used to provide a field in the direction of θ_{SOI} is to ensure that the fields radiated from elements A_{-2}, A_{-1}, A_1 and A_2 arrive

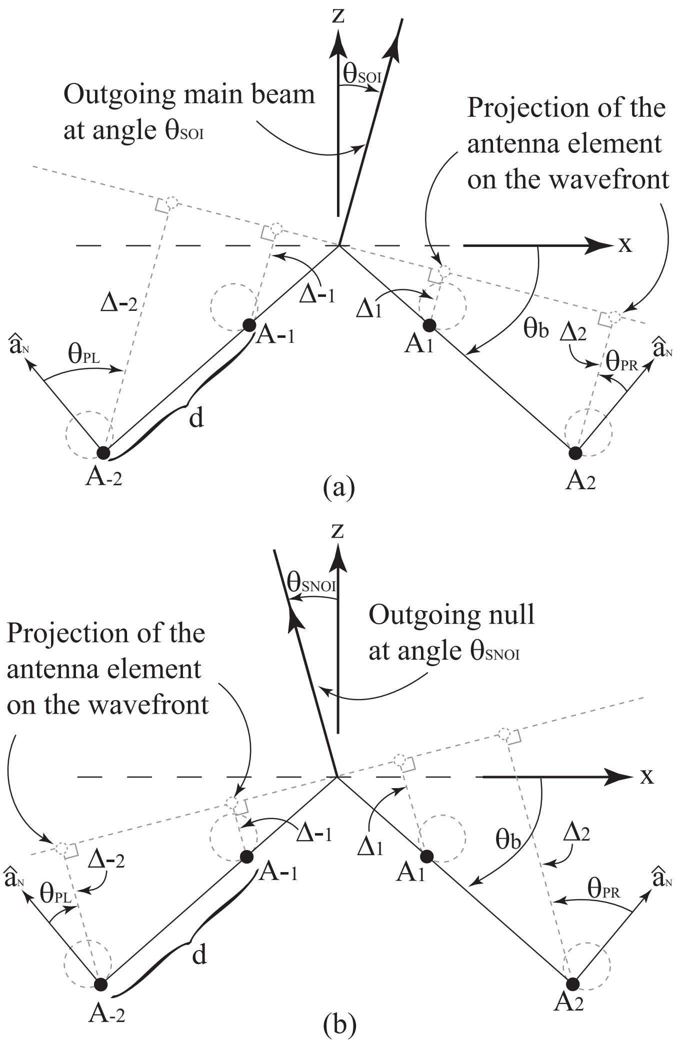

Figure 17. (a) Illustration of the 1 × 4 conformal antenna on a wedge-shaped surface with a transmitted signal of interest at angle θ_{SOI} and (b) illustration of the 1 × 4 conformal antenna on a wedge-shaped surface with a transmitted signal not of interest at angle θ_{SNOI}.

at the wavefront with the same phase. This will then result in a broad-side radiation to the wavefront. To ensure that these fields arrive with the same phase, the voltage phase driving each individual element can be adjusted with a phase shifter. The amount of phase required is equal and opposite to sign of the phase introduced by the free-space propagation from the antenna element to the wavefront. To compute this distance, the values for Δ_{-2}, Δ_{-1}, Δ_1 and Δ_2 in Figure 17(a), which denote the distance from the elements A_{-2}, A_{-1}, A_1 and A_2, respectively, to the wavefront (or the projected elements on the wavefront), need to be computed. Furthermore, the expressions for A_{-2}, A_{-1}, A_1 and A_2 should be written in a general manner that includes the location of each antenna element on the surface and the bend angle θ_b. A similar argument can be made for the case when the array is transmitting a null in the direction of θ_{SNOI_n}, which is shown in Figure 17(b). For this case, the values of $\Delta_{\pm n}$ still need to be computed.

3.1. Computing the distance to the projected element

For this work, the values of θ_{SOI} and θ_{SNOI_n} are between $-\pi/2$ and $\pi/2$. Because of this, the problem will be broken down in to two cases. For the first case, $-\theta_b \leq \theta_{SOI(SNOI_n)} \leq \theta_b$. For these angles of $\theta_{SOI(SNOI_n)}$ the projected elements on the wavefront are all outside of the wedge-shaped surface. For the second case, $\theta_b \leq \theta_{SOI(SNOI_n)} \leq \pi/2$ or $-\pi/2 \leq \theta_{SOI(SNOI_n)} \leq -\theta_b$. For these angles of $\theta_{SOI(SNOI_n)}$, half of the projected elements are outside of the wedge-shaped surface and half are inside of the surface. This is because when $\theta_{SOI(SNOI_n)} \geq \theta_b$ or $\theta_{SOI(SNOI_n)} \leq -\theta_b$ the projected elements for A_1 and A_2 or A_{-1} and A_{-2}, respectively, are inside of the wedge-shaped surface in Figure 17.

1) Case 1: computing $\Delta_{\pm n}$ for $-\theta_b \leq \theta_{SOI(SNOI_n)} \leq \theta_b$:

Using the notation in Figure 17, the distance from the antenna elements on the wedge to the projected elements on the wavefront can be computed using:

$$\Delta_{\pm n} = d\frac{|n|}{2n}(2|n| - 1)\sin\left(\theta_b \mp \theta_{SOI(SNOI_n)}\right) \tag{3.1}$$

for $0 \leq \theta_{SOI(SNOI_n)} \leq \theta_b$ and

$$\Delta_{\pm n} = d\frac{|n|}{2n}(2|n| - 1)\sin\left(\theta_b \pm |\theta_{SOI(SNOI_n)}|\right) \tag{3.2}$$

for $-\theta_b \leq \theta_{SOI(SNOI_n)} \leq 0$. Notice that the expressions in (3.1) and (3.2) are written in terms of the bend angle θ_b of the wedge and the location of each antenna element, indicating the generality of the expressions.

2) Case 2: computing $\Delta_{\pm n}$ for $\theta_b \leq \theta_{SOI(SNOI_n)} \leq \pi/2$ or $-\pi/2 \leq \theta_{SOI(SNOI_n)} \leq -\theta_b$:

Next, for the remaining values of $\theta_{SOI(SNOI_n)}$ the distance from each element in the array to the projected elements on the wavefront can be computed as:

$$\Delta_{\pm n} = d\frac{|n|}{2n}(2|n| - 1)\sin\left(\theta_{SOI(SNOI_n)} \mp \theta_b\right) \tag{3.3}$$

for $\theta_b \leq \theta_{SOI(SNOI_n)} \leq \pi/2$ and

$$\Delta_{\pm n} = d\frac{|n|}{2n}(2|n| - 1)\sin\left(|\theta_{SOI(SNOI_n)}| \pm \theta_b\right) \tag{3.4}$$

for $-\pi/2 \leq \theta_{SOI(SNOI_n)} \leq -\theta_b$. Notice that (3.3) and (3.4) are also written in terms of θ_b and the position of each antenna element in a general fashion.

3.2. Computing the radiation in the direction of θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}

Next, the field from the array in the direction of θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3} on the wedge-shaped surface can be computed using the following array factor (AF_w)

expression [55]:

$$AF_w(\theta,\phi) = \sum_{n=-2}^{-1} F_{n,L} e^{jk[ux_n+vy_n+z_n\cos\theta]} + \sum_{n=1}^{2} F_{n,R} e^{jk[ux_n+vy_n+z_n\cos\theta]} \tag{3.5}$$

where

$$F_{n,L} = w_n \cos(\theta_{PL}) e^{\pm jk\Delta_{\pm n}} \tag{3.6}$$

and

$$F_{n,R} = w_n \cos(\theta_{PR}) e^{\pm jk\Delta_{\pm n}}. \tag{3.7}$$

A spherical coordinate system is assumed in (3.5) with $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$, $\Delta_{\pm n}$ is defined in (3.1)-(3.4), $(x_n,\ y_n,\ z_n)$ is the location of the n^{th} array element and w_n are the complex weighting functions (i.e., array weights). Furthermore, the element patterns for A_1 and A_2 are denoted as $e_R(\theta) = \cos\theta_{PR}$ and the element patterns for A_{-1} and A_{-2} are denoted as $e_L(\theta) = \cos\theta_{PL}$ where θ_{PR} and θ_{PL} are defined in Figure 17. To compute the complex excitation coefficients for antenna elements on the conformal wedge we determine θ_{PL} and θ_{PR} as shown in Figure 17. These are the offset directions of SOI and SNOIs on the conformal wedge for the left and right antenna elements of the array midpoint and must be included to compute the weighting coefficients to give the desired SOI and SNOIs radiation pattern. To compute θ_{PL} and θ_{PR} for $\theta_{SOI(SNOI)} > 0$, consider Figure 17(a). From the geometry, we can write

$$\theta_{PL} = (\theta_b + |\theta|) \tag{3.8}$$

and

$$\theta_{PR} = -(\pi/2 - |\theta| - (\pi/2 - \theta_b)) = -\theta_b + |\theta|. \tag{3.9}$$

Similary for $\theta_{SOI(SNOI)} < 0$ in Figure 17(b) we can write

$$\theta_{PL} = \pi/2 - [(\pi/2 - \theta_b) + |\theta|] = \theta_b - |\theta| \tag{3.10}$$

and

$$\theta_{PR} = -(\theta_b + |\theta|) = -\theta_b - |\theta|. \tag{3.11}$$

3.3. Computing the array weights for $N = 4$ elements

The complex weighting functions (i.e., array weights) in (3.5) are of interest in this work because these are the array weights that need to be computed for the beamforming of the array. To compute these weights, the matrix method for smart antennas presented in [71] will be used. Because the array studied here has four-elements, four array weights are needed. This then requires a set of four equations with four unknowns that can be written as a square matrix. Then the array weights can be computed using a matrix solver. The sum in (3.5) has four terms for $N = 4$ and to get four equations with four unknowns, (3.5) will be evaluated at each value of θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}. This then gives the following array factor matrix $\mathbf{AF_w}$:

$$\mathbf{AF_w} = \begin{bmatrix} AF_w(\theta_{SOI}) \\ AF_w(\theta_{SNOI_1}) \\ AF_w(\theta_{SNOI_2}) \\ AF_w(\theta_{SNOI_3}) \end{bmatrix} \tag{3.12}$$

where $\theta = \theta_{SOI}$, θ_{SNOI_1}, θ_{SNOI_2} or θ_{SNOI_3} and $\phi = 0$. Then, factoring out the array weights gives

$$\mathbf{AF_w} = \mathbf{AW} \tag{3.13}$$

where $\mathbf{A}$ is the array factor on the wedge in (3.5) with the weights factored out and

$$\mathbf{W} = \begin{bmatrix} w_{-2} \\ w_{-1} \\ w_{1} \\ w_{2} \end{bmatrix}. \tag{3.14}$$

Next, to ensure that the conformal antenna will have a main beam at the scan angle θ_{SOI}, $AF_w(\theta_{SOI})$ must equal 1 in (3.12), or $AF_w(\theta_{SOI})$=1. Then, in order to provide nulls in the directions of θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}, $AF_w(\theta_{SNOI_1})$=$AF_w(\theta_{SNOI_2})$=$AF_w(\theta_{SNOI_3})$=0 in (3.12). This can be written in matrix form in the following manner:

$$\mathbf{AW} = \mathbf{C} \tag{3.15}$$

where

$$\mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.16}$$

and $\mathbf{AW}$ is defined in (3.13). Finally, solving for the weights in (3.15) gives

$$\mathbf{W} = \mathbf{A^{-1}C}. \tag{3.17}$$

By setting the first element in (3.16) to be 1 and the rest of the elements to be 0, the array factor is forced to give nulls (or a zero field) analytically at the angles of θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}. Thus, the solution of (3.17) yields the array weights required to give these pattern null features. Furthermore, since the values of $\Delta_{\pm n}$ in the array factor expression in (3.12) are written in terms of the wedge angle θ_b and the location of each antenna element, the weights computed with (3.17) are determined in a setting where the antenna, and hence the wedge, can change shape.

3.4. Computing the array weights for N elements

The previous expressions were determined for $N = 4$ and can be generalized for more elements. More specifically, equations (3.12)-(3.17) can be generalized in the following manner:

$$\mathbf{AF_w} = \begin{bmatrix} AF_w(\theta_{SOI}) \\ \vdots \\ AF_w(\theta_{SNOI_1}) \\ \vdots \\ AF_w(\theta_{SNOI_n}) \end{bmatrix} \tag{3.18}$$

where

$$\begin{aligned} AF_{wn}(\theta, \phi) =& \sum_{n=-N/2}^{-1} F_{n,L} e^{jk[ux_n + vy_n + z_n \cos\theta]} \\ &+ \sum_{n=1}^{N/2} F_{n,R} e^{jk[ux_n + vy_n + z_n \cos\theta]}, \end{aligned} \tag{3.19}$$

and $F_{n,L}$ and $F_{n,R}$ are defined in (3.6) and (3.7), respectively. Then, factoring out the array weights gives

$$\mathbf{AF_{wn}} = \mathbf{A_n W_n} \tag{3.20}$$

where $\mathbf{A}$ is the array factor on the wedge in (3.19) with the weights factored out and

$$\mathbf{W_n} = \begin{bmatrix} w_{-N/2} \\ \vdots \\ w_{-1} \\ w_1 \\ \vdots \\ w_{N/2} \end{bmatrix}. \tag{3.21}$$

Finally, the coefficients in (3.16) can be generalized as

$$\mathbf{C_n} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \tag{3.22}$$

and the weights can be computed using

$$\mathbf{W_n} = \mathbf{A_n^{-1}} \mathbf{C_n}. \tag{3.23}$$

Equation (3.23) can then be used to solve for the array weights that will result in a main radiation pattern in the direction of θ_{SOI} and nulls in the directions of θ_{SNOI_1}, θ_{SNOI_2}, ... $\theta_{SNOI_{(N-1)}}$ for an N-element array on a changing wedge-shaped surface.

CHAPTER 4. BEAMFORMING OF A 1×4 ARRAY ON A CYLINDRICAL-SHAPED CONFORMAL SURFACE

The array weights for the conformal antenna on the cylindrical shaped surface shown in Figure 18 are computed next. Again, for illustration, the derivations will be carried out for $N = 4$ elements.

4.1. Computing the distance to the projected element

As with the wedge-shaped conformal surface, the distance from the elements on the cylindrical surface to the projected elements on the transmitted wavefront at angle $\theta_{SOI(SNOI_n)}$ must be computed. Again, these distances are denoted as $\Delta_{\pm n}$ in Figure 18. First, the distance from the point where the cylinder intersects the z-axis (denoted as point P in Figure 18) to each antenna element is computed using

$$h_{\pm n} = \sqrt{(0 - x_{\pm n})^2 + (r - z_{\pm n})^2} \tag{4.1}$$

where again $(x_{\pm n}, z_{\pm n})$ is the location of the n^{th} element on the cylindrical surface, $x_{\pm n} = r \cos \phi_{\pm n}$, $z_n = r \sin \phi_{\pm n}$ and $\phi_{\pm n}$ is defined in Figure 18. As with the wedge-shaped surface, the problem will be considered as two different cases. The first case is for $\theta_{SOI(SNOI_n)} \geq 0$ and the second case is for $\theta_{SOI(SNOI_n)} \leq 0$. The first case is shown in Figure 18. The projected elements of A_{-2}, A_{-1} and A_2 in the direction of $\theta_{SOI(SNOI_n)}$ are outside of the cylindrical surface and the projected element of A_1 in the direction of $\theta_{SOI(SNOI_n)}$ is inside of the cylindrical surface. Because some projected elements are outside of the cylinder and some are inside, each case will be broken into two parts. The first part will compute the distance from the elements A_{-1} and $A_{\pm 2}$ to the projected elements on the wavefront and the second part will compute the distance from the element A_1 to the projected element on the wavefront (as shown in Figure 18).

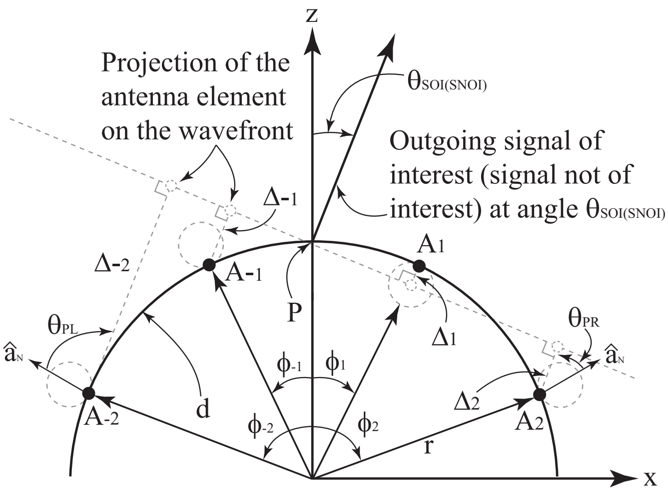

Figure 18. Illustration of the 1 × 4 conformal antenna on a cylindrical-shaped surface with a transmitted signal of interest (signal not of interest) at angle $\theta_{SOI(SNOI)}$.

1) Case 1 with $\theta_{SOI(SNOI_n)} \geq 0$

Next, using (4.1) the distance from the elements A_{-1} and $A_{\pm 2}$ to the projected elements on the transmitted wavefront can be computed as:

$$\Delta_{\pm n} = h_{\pm n} \sin(\mp\theta_{SOI(SNOI_n)} + \theta_{\pm n}) \tag{4.2}$$

where $\theta_{\pm n} = \cos^{-1}|x_{\pm n}/h_{\pm n}|$. Then, for element A_1 the distance to the projected element can be computed as:

$$\Delta_{+n} = h_{+n} \sin(\theta_{SOI(SNOI_n)} - \theta_{+n}). \tag{4.3}$$

2) Case 2 with $\theta_{SOI(SNOI_n)} \leq 0$

Again, using (4.1) the distance from the elements A_1 and $A_{\pm 2}$ to the projected elements can be computed as:

$$\Delta_{\pm n} = h_{\pm n} \sin(\pm|\theta_{SOI(SNOI_n)}| + \theta_{\pm n}) \tag{4.4}$$

where $\theta_{\pm n} = \cos^{-1}|x_{\pm n}/h_{\pm n}|$. Then for element A_{-1} the distance to the projected element can be computed as:

$$\Delta_{-n} = h_{-n}\sin(|\theta_{SOI(SNOI_n)}| - \theta_{-n}). \tag{4.5}$$

A few comments can be made about (4.2)-(4.5). As with the wedge-shaped surface, these expressions have been written in a general manner in terms of antenna position on the cylindrical surface and the radius of curvature. Also, special care should be taken when implementing these equations if $|\theta_{SOI(SNOI_n)}| \leq \theta_{\pm 1}$. This is because for these angles all of the projected elements on the wavefront will be outside of the cylindrical surface. In this case, the distance to the wavefront can be computed using (4.2) or (4.4) (depending on whether $\theta_{SOI(SNOI_n)} \leq 0$ or $\theta_{SOI(SNOI_n)} \geq 0$) and the computations of (4.3) and (4.5) are not required.

4.2. Computing the radiation in the direction of θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3}

Next, the field from the array in the direction of θ_{SOI}, θ_{SNOI_1}, θ_{SNOI_2} and θ_{SNOI_3} on the cylindrical-shaped surface can be computed using (3.5) (with updated distances computed using (4.2)-(4.5). As mentioned in Section 3.2, Chapter 3, to obtain the correct SOI and SNOIs radiation pattern for conformal antennas array, we need to determine θ_{PL} and θ_{PR}. To compute θ_{PL} and θ_{PR} for $\theta_{SOI(SNOI)} < 0$, consider Figure 18. From the geometry, we can write

$$\theta_{PL} = \pi/2 - \phi_n - |\theta| \tag{4.6}$$

and

$$\theta_{PR} = -(\pi/2 - \phi_n + |\theta|). \tag{4.7}$$

Similary for $\theta_{SOI(SNOI)} > 0$ in Figure 18 we can write

$$\theta_{PL} = \pi/2 - \phi_n + |\theta| \tag{4.8}$$

and

$$\theta_{PR} = -(\pi/2 - \phi_n - |\theta|) = -\pi/2 + (\phi_n + |\theta|. \tag{4.9}$$

4.3. Computing the array weights for $N = 4$ elements

Next, as with the conformal antenna on the wedge-shaped surface, the array weights can be computed using (3.17). For the computations of these array weights on the cylinder, the distance from the elements to the wavefront should be computed using (4.2)-(4.5).

4.4. Computing the array weights for N elements

Finally, (3.23) can be used (with updated distances computed using (4.2)-(4.5)) to compute the array weights for 1×N array on a cylindrical surface with radius r. As with the wedge-shaped surface, the expressions to compute the array weights have been written in a general manner that includes element spacing and the radius of the cylinder. This makes the technique presented here useful for an array attached to a changing cylindrical surface.

CHAPTER 5. COMPUTING THE WEIGHTING COEFFICIENTS WITH MUTUAL COUPLING

For the previous derivations of (3.17) and (3.23), it was assumed that there was no mutual coupling between the elements. To model the mutual coupling, the work presented in [93] was considered. More specifically, the work in [93] proposed a model for the mutual coupling in adaptive arrays and demonstrated that the mutual coupling between the elements of an adaptive array can cause a significant degradation in the signal-to-interference-plus-noise ratio (SINR). The methods in [93] will be adopted here to model the mutual coupling effects on the radiation pattern of the conformal beamforming array.

The coupling between the antenna elements can be modeled as an N+1 port network, as shown in Figure 19. The antenna elements in the array are all terminated with Z_L and denoted as ports 1, 2, ... N. The antenna port being driven by a voltage source V_s is denoted as N+1. The port being driven with V_s is the representation of the transmitted (or incoming) signals at angles θ_{SOI} or θ_{SNOI_n}. Then, using the Kirchhoff relations for the $N+1$ terminal network, the voltage at the terminated port n can be written as:

$$V_{Tn} = I_1 Z_{n,1} + I_2 Z_{n,2} + .. + I_n Z_{n,m} + ... + I_s Z_{n,s} \tag{5.1}$$

where $Z_{n,m}$ is the mutual impedance between the n^{th} and m^{th} port, I_n is the current going through the terminating resistor on the n^{th} port, Z_{nn} is the self-impedance of the n^{th} port and $Z_{n,s}$ represents the mutual coupling term between the driven element with V_s and the n^{th} antenna element.

Furthermore, the current at the n^{th} port can be written in terms of the terminal

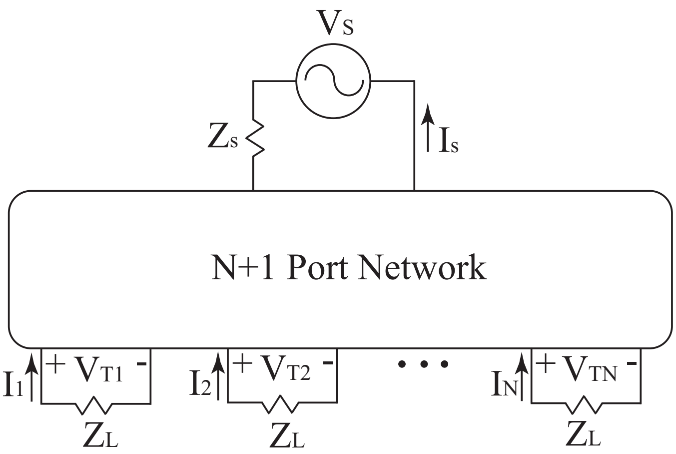

Figure 19. N-port network illustration of the conformal array with a signal of interest at angle θ_{SOI} or θ_{SNOI_n}.

voltage and load impedance in the following manner:

$$I_n = \frac{-V_{Tn}}{Z_L}. \tag{5.2}$$

Then, making use of the open-circuit condition and removing the terminating impedances results in $I_1 = I_2 = ... = I_N = 0$. This then simplifies (5.1) to $V_{Tn} = I_s Z_{n,s}$. Under these conditions, (5.1) represents the open circuit voltages at the n^{th} port caused by the mutual coupling between the driven element and the n^{th} port, and can be computed as $V_{Tn} = I_s Z_{n,s} = V_{OCn}$. Next, substituting (5.2) into (5.1), making use of the open-circuit condition and writing (5.1) in matrix form results in the following:

$$\mathbf{Z_c V_T} = \mathbf{V_{OC}}. \tag{5.3}$$

where

$$\mathbf{Z_c} = \begin{bmatrix} 1 + \frac{Z_{1,1}}{Z_L} & \frac{Z_{1,2}}{Z_L} & \dots & \frac{Z_{1,N}}{Z_L} \\ \frac{Z_{2,1}}{Z_L} & 1 + \frac{Z_{2,2}}{Z_L} & \dots & \frac{Z_{2,N}}{Z_L} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{Z_{N,1}}{Z_L} & \frac{Z_{N,2}}{Z_L} & \dots & 1 + \frac{Z_{N,N}}{Z_L} \end{bmatrix}, \tag{5.4}$$

$$\mathbf{V_T} = \begin{bmatrix} V_{T1} \\ V_{T2} \\ \vdots \\ V_{Tn} \end{bmatrix} \tag{5.5}$$

and

$$\mathbf{V_{OC}} = \begin{bmatrix} V_{OC1} \\ V_{OC2} \\ \vdots \\ V_{OCn} \end{bmatrix}. \tag{5.6}$$

The normalized impedance matrix $\mathbf{Z_c}$ includes self- and mutual-terms, and can be determined from a 3D full wave simulator such as HFSS [94]. The open circuit voltage column matrix $\mathbf{V_{OC}}$ represents the array weights (i.e., the complex voltages used to drive the beamforming conformal antenna) without including mutual coupling. This then results in $\mathbf{V_{OC}} = \mathbf{W_n}$ where $\mathbf{W_n}$ is given in (3.21). The terminal voltage column matrix $\mathbf{V_T}$ represents the array weights that include the mutual coupling and can be computed from (5.3) as follows:

$$\mathbf{V_T} = \mathbf{Z_c^{-1}} \mathbf{V_{OC}}. \tag{5.7}$$

Next, to write the array weights with mutual coupling in terms of the array factor expressions in (3.20), (3.20) is equated to (3.22), and $\mathbf{V_{OC}} = \mathbf{W_n}$ is substituted into the equality. This results in the following expression:

$$\mathbf{A_n} \mathbf{V_{OC}} = \mathbf{C_n}. \tag{5.8}$$

Then, solving for $\mathbf{V_{OC}}$ in (5.7), substituting into (5.8) and reorganizing gives:

$$\mathbf{V_T} = \mathbf{Z_c^{-1}} \mathbf{A_n^{-1}} \mathbf{C_n} = \mathbf{W_n^c} \tag{5.9}$$

where $\mathbf{W_n^c}$ represents the new array weights with the coupling included in the computations. In the next section, validation of $\mathbf{W_n}$ and $\mathbf{W_n^c}$ for various values of θ_{SOI} and θ_{SNOI_n} will be presented followed by the characteristics of the array weights for various conformal surfaces.

CHAPTER 6. DEPENDENCE OF THE ARRAY WEIGHTS ON THE CONFORMAL SURFACE GEOMETRY

In this chapter, the array weights for a three-element array were computed analytically using (3.23). This was done to show how the array weights are dependent on θ_b and the radius of curvature r. It should be mentioned that $N = 3$ was chosen because the problem becomes much more complex for $N > 3$ and requires a numerical analysis of the matrices in (3.23). Assuming one SOI and two SNOIs, we can write the following three equations with three unknown array weights:

$$w_{-1}F_{-1,L}(\theta_{SOI}) + w_0 + w_{+1}F_{1,R}(\theta_{SOI}) = 1, \tag{6.1}$$

$$w_{-1}F_{-1,L}(\theta_{SNOI_1}) + w_0 + w_{+1}F_{1,R}(\theta_{SNOI_1}) = 0, \tag{6.2}$$

and

$$w_{-1}F_{-1,L}(\theta_{SNOI_2}) + w_0 + w_{+1}F_{1,R}(\theta_{SNOI_2}) = 0. \tag{6.3}$$

Subtracting (6.2) and (6.3) from (6.1) we obtain

$$w_{-1}[F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_1})] + w_{+1}[F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_1})] = 1, \tag{6.4}$$

$$w_{-1}[F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_2})] + w_{+1}[F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_2})] = 1, \tag{6.5}$$

Next subtracting (6.3) from (6.2) we obtain

$$w_{-1}[F_{-1,L}(\theta_{SONI_1}) - F_{-1,L}(\theta_{SNOI_2})] + w_{+1}[F_{1,R}(\theta_{SNOI_1}) - F_{1,R}(\theta_{SNOI_2})] = 0. \quad (6.6)$$

Equation (6.6) can be written as:

$$w_{-1} = -w_{+1}\frac{F_{1,R}(\theta_{SNOI_2}) - F_{1,R}(\theta_{SNOI_1})}{F_{-1,L}(\theta_{SONI_2}) - F_{-1,L}(\theta_{SNOI_1})}, \quad (6.7)$$

or

$$w_{-1} = -w_{+1}D \quad (6.8)$$

where

$$D = \frac{F_{1,R}(\theta_{SNOI_2}) - F_{1,R}(\theta_{SNOI_1})}{F_{-1,L}(\theta_{SNOI_2}) - F_{-1,L}(\theta_{SNOI_1})}. \quad (6.9)$$

Next, substituting (6.8) into (6.4) we get

$$w_{+1}[F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_1})] = 1 + w_{+1}D[F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_1})], \quad (6.10)$$

or

$$w_{+1}[F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_1})] - D[F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_1})] = 1., \quad (6.11)$$

Now, from (6.11) we can calculate w_{+1} as follows:

$$w_{+1} = \frac{1}{[F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_1})] - D[F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_1})]} \quad (6.12)$$

or

$$w_{+1} = \frac{1}{F^{\Delta}_{1,R} - DF^{\Delta}_{-1,L}}. \quad (6.13)$$

Equation (6.13) can be used to calculate w_{-1} from (6.8). Finally using (6.8) and

(6.13) in (6.1), we can calculate the array weight w_0 as follows:

$$w_0 = 1 - w_{+1}[F_{1,R}(\theta_{SOI}) - DF_{-1,L}(\theta_{SOI})], \tag{6.14}$$

where

$$D = \frac{F_{1,R}(\theta_{SNOI_2}) - F_{1,R}(\theta_{SNOI_1})}{F_{-1,L}(\theta_{SNOI_2}) - F_{-1,L}(\theta_{SNOI_1})}. \tag{6.15}$$

Also, $F^{\Delta}_{1,R} = F_{1,R}(\theta_{SOI}) - F_{1,R}(\theta_{SNOI_1})$, $F^{\Delta}_{-1,L} = F_{-1,L}(\theta_{SOI}) - F_{-1,L}(\theta_{SNOI_1})$ and $F_{n,L}$ and $F_{n,R}$ are defined in (3.6) and (3.7), respectively, with the weights factored out.

For a small value of N, the results in (6.8), (6.13) and (6.14) show that the expression for the array weights can be quite complicated. Then again, it does also show the dependence of each weight on the angle $\theta_{SOI(SNOI_n)}$ and the location of each antenna element on the array.

To illustrate the behavior of the weights computed using (6.8), (6.13) and (6.14) for various surfaces, the three element array was considered on a wedge-shaped surface for various values of θ_b. For these computations the values of θ_{SOI} and θ_{SNOI_n} were $\theta_{SOI} = 40°$, $\theta_{SNOI_1} = -45°$ and $\theta_{SNOI_2} = 10°$. The amplitude and phase of the array weights computed using (6.8), (6.13) and (6.14) are shown in Figs. 20 and 21, respectively for wedge-shaped conformal surface. Similarly the amplitude and phase of the array weights for various values of radius of curvature on a cylindrical surface are shown in Figs. 22 and 23, respectively. The results show that for a fixed beamformation there is a strong relationship between the array weights and the conformal surface.

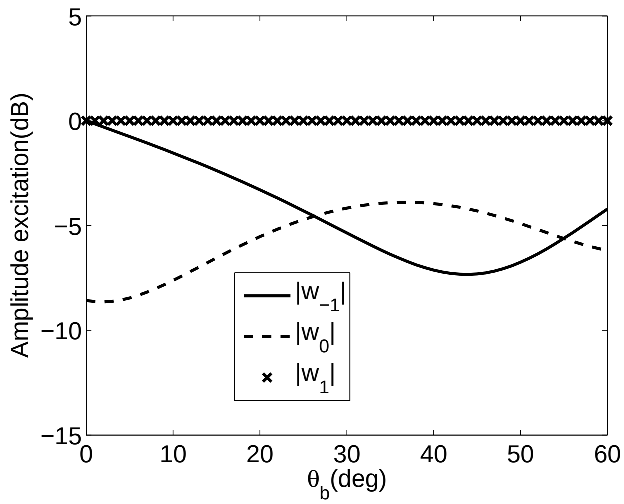

Figure 20. Magnitude of the array weights for the three-element array on the wedge-shaped surface.

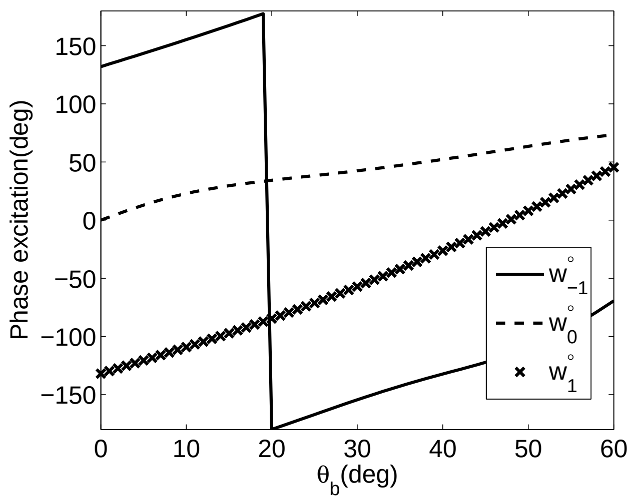

Figure 21. Phase of the array weights for the three-element array on the wedge-shaped surface.

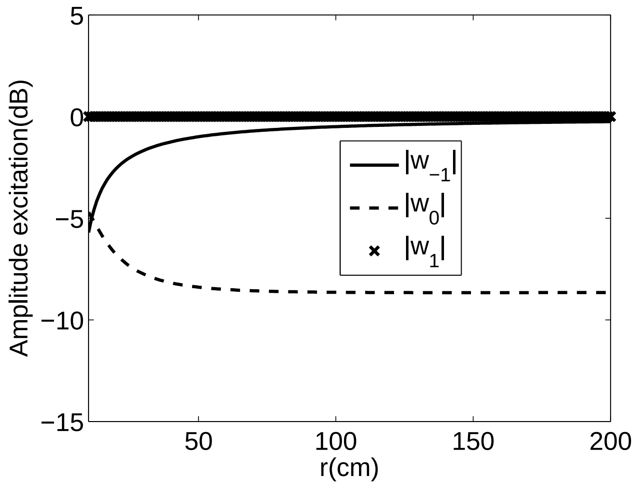

Figure 22. Magnitude of the array weights for the three-element array on the cylindrical-shaped surface.

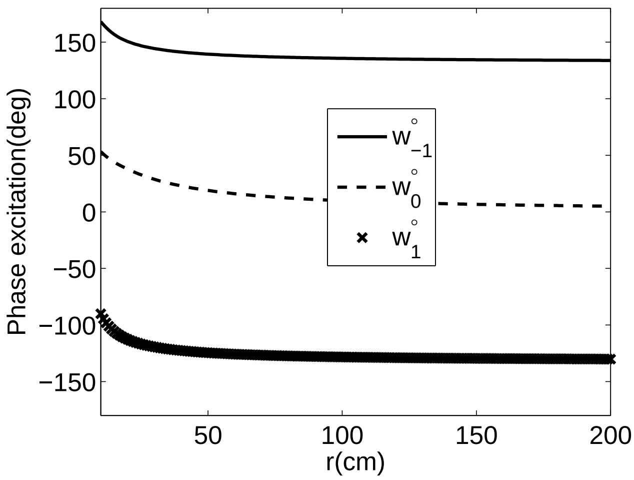

Figure 23. Phase of the array weights for the three-element array on the cylindrical-shaped surface.

CHAPTER 7. VALIDATION WITH ANALYTICAL, SIMULATION AND MEASUREMENT RESULTS

In this chapter, a beamforming array is used to validate the previously derived array weight expressions using simulations and measurements. More specifically the array weights are computed using (3.23) and (5.9) for the four-element array on the wedge- and cylindrical-shaped surface shown in Figs. 17 and 18. Two different beamformation patterns were considered and each pattern was evaluated on three different conformal surfaces. The characteristics of each beamformation pattern, denoted as pattern 1 and pattern 2, are summarized in Table 1. Furthermore, the three conformal surfaces considered were the wedge-shaped surface with $\theta_b = 30°$ and $\theta_b = 45°$ and a cylindrical surface with $r = 10$ cm.

7.1. The four-element beamforming array prototype

For measurement purposes, the four-element beamforming array shown in Figure 24(a) was manufactured. The array consisted of connectorized voltage-controlled phase shifters, voltage-controlled attenuators, a four-way power divider, an amplifier and four microstrip patches designed to operate at 2.47 GHz. A picture of the attenuators, phase shifters and power divider of the manufactured array is shown in Figure 24(b) and a picture of the four microstrip patches is shown in Figure 24(c). Four individual microstrip patches were used for the convenience of placing the array on the various conformal surfaces. The phase shifters were manufactured by Hittite Microwave Corporation [95] (PN: HMC928LP5E) and the power divider, attenuators and amplifiers were manufactured by Mini-Circuits [96] (PNs: ZN8PD1-53-S+, ZX73-2500-S+ and ZX60-33LN-S+, respectively). Identical SMA cables were used to connect each patch to a port on the power divider. The attenuation and phase shift of the voltage variable attenuator and anlog phase shifter were measured and are shown in Tables 8 and 9 at the end of the chapter.

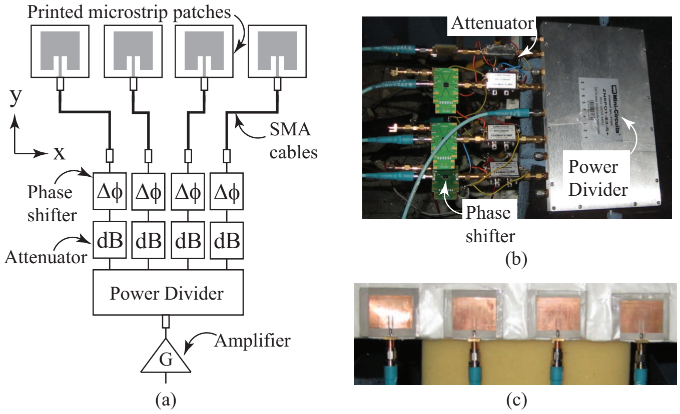

Figure 24. (a) Topology of the four-element beamforming array, (b) a photograph of the power divider, voltage controlled phase shifters and voltage controlled attenuators used for measurements and (c) a photograph of the microstrip patch elements used for attachment to conformal surfaces.

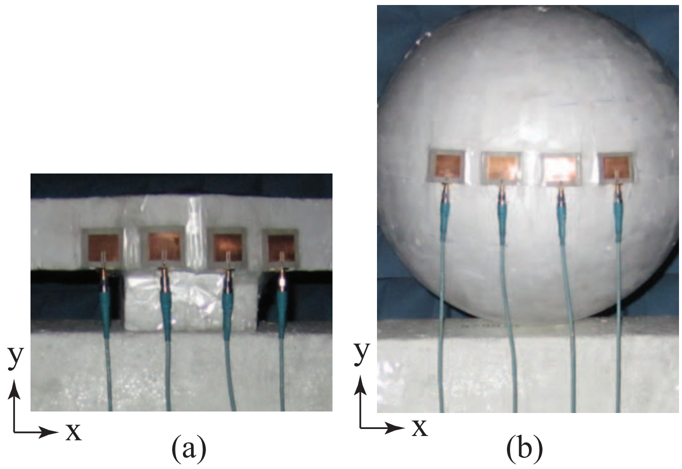

Figure 25. (a) Photograph of the four-element beamforming array being measured on the wedge-shaped surface with $\theta_b = 30°$ and (b) photograph of the four-element beamforming array being measured on the cylindrical shaped surface with $r = 10$ cm.

Table 1. Summary of the beamformation patterns 1 and 2.

Variable	Pattern 1	Pattern 2
θ_{SOI}	0°	40°
θ_{SNOI_1}	−30°	−45°
θ_{SNOI_2}	30°	−25°
θ_{SNOI_3}	40°	10°

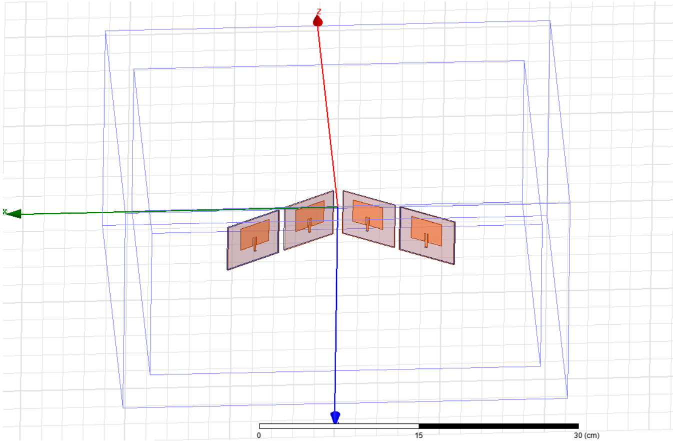

Figure 26. Picture of the four-element beamforming array in HFSS.

7.2. Beamforming results on the wedge with $\theta_b = 30°$

The first measurements taken were for the four-element beamforming array on the wedge-shaped surface in Figure 17 for $\theta_b = 30°$ and the array weights were computed using (3.17). The inter-element spacing was 0.5 λ and a picture of the array being measured on the surface with a 2-port network analyzer in a fully calibrated anechoic chamber is shown in Figure 25(a). The results from these measurements for both patterns summarized in Table 1 are shown in Figures 27 and 28. Next, the four-element beamforming array was simulated in HFSS as shown in Figure 26 and the weights computed using (3.17) were used to drive the array. The radiation pattern from these simulations can also be seen in Figures 27 and 28 at 2.47 GHz. Then, for a third comparison both patterns were computed analytically using the array factor expression in (3.5) with the weights determined using (3.17). These results are also shown in Figures 27 and 28. Finally, new weights that include the mutual coupling were computed using (5.9) and used in (3.5) to compute the radiation pattern analytically. These results are shown in Figures 27 and 28. Overall, agreement between measurements, simulations and analytical computations (with both sets of array weights) is shown. The array weights for the results in Figures 27 and 28 are shown in Tables 2 and 3 at the end of the chapter.

7.3. Beamforming results on the wedge with $\theta_b = 45°$

Next, measurements were taken for the four-element beamforming array on the wedge-shaped surface in Figure 17 for $\theta_b = 45°$. The inter-element spacing was again 0.5 λ and (3.17) was used to compute the new array weights. The results from these measurements for both patterns summarized in Table 1 are shown in Figures 29 and 30. Next, the four-element beamforming array was simulated in HFSS with the weights for $\theta_b = 45°$ and the radiation pattern can also be seen in Figures 29 and 30 at 2.47 GHz. Then, for a third comparison the expression in (3.5) was used to compute the analytical results shown in Figures 29 and 30. New weights that include the mutual coupling were also computed using (5.9) for the new value of $\theta_b = 45°$. These weights were then used in (3.5) to compute the radiation pattern and these results are shown in Figures 29 and 30. As with the $\theta_b = 30°$ results, agreement between measurements, simulations and analytical computations is shown. The array weights for the results in Figures 29 and 30 are also shown in Tables 4 and 5 at the end of the chapter.

7.4. Beamforming results on the cylinder with $r = 10$ cm

Finally, the same comparison between measurements, simulations and analytical computations was conducted for the four-element array on the cylindrical-shaped surface with $r = 10$ cm. The array being measured in the full anechoic chamber is shown in Figure 25(b) and the results are shown in Figures 31 and 32. It should be mentioned that in order to measure the array on a cylindrical surface a sphere was used and the antenna elements were placed along the equator. This then resulted in an antenna shape similar to a cylindrical surface. Simulation results and analytical computations with and without the coupling are also shown in Figures 31 and 32 and overall agreement is shown. The array weights for the results in Figures 31 and 32 are shown in Tables 6 and 7 at the end of the chapter.

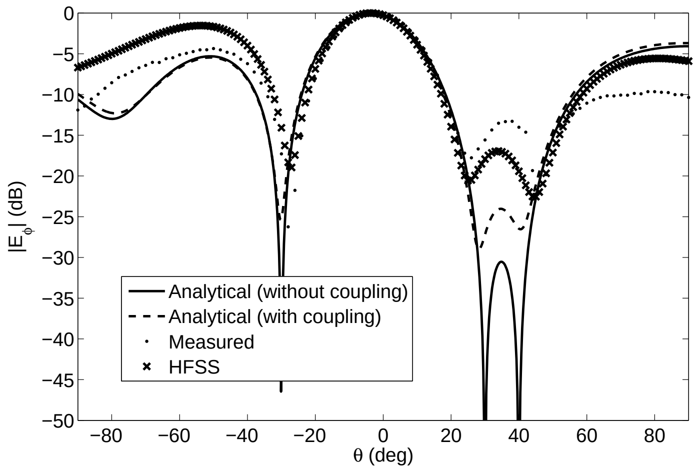

Figure 27. Pattern 1 beamforming results for the four-element array on the wedge-shaped surface with $\theta_b = 30°$.

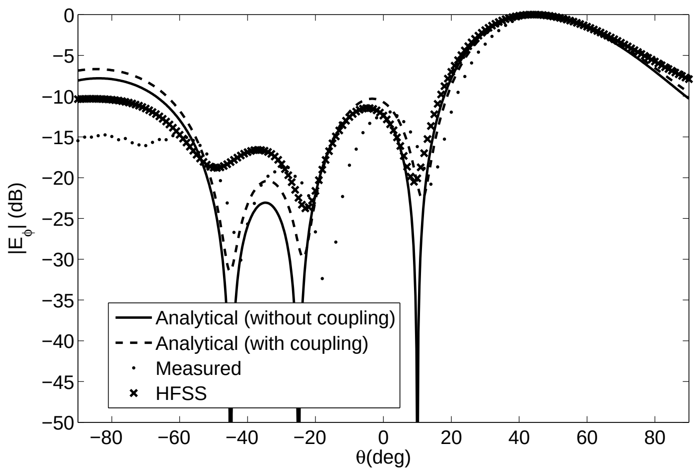

Figure 28. Pattern 2 beamforming results for the four-element array on the wedge-shaped surface with $\theta_b = 30°$.

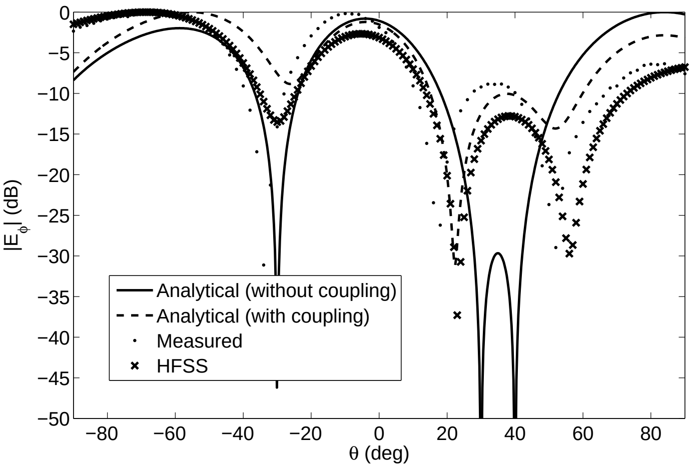

Figure 29. Pattern 1 beamforming results for the four-element array on the wedge-shaped surface with $\theta_b = 45°$.

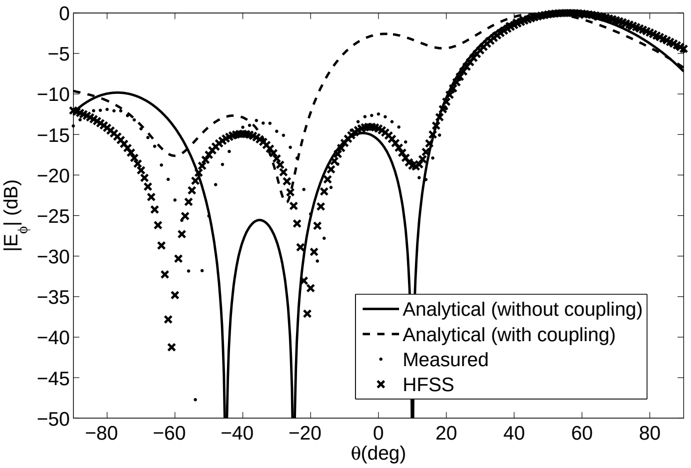

Figure 30. Pattern 2 beamforming results for the four-element array on the wedge-shaped surface with $\theta_b = 45°$.

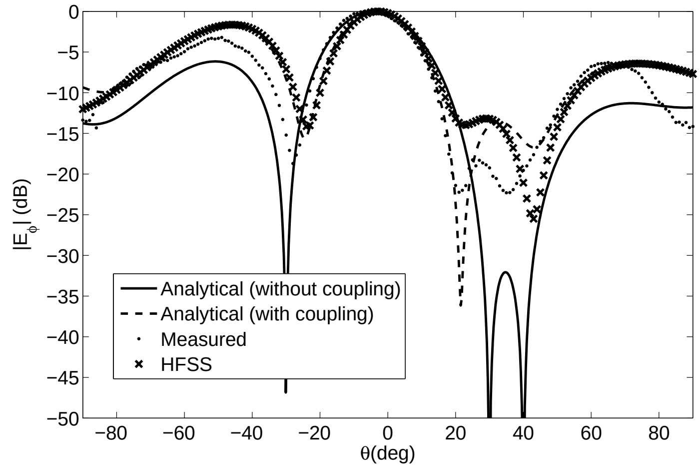

Figure 31. Pattern 1 beamforming results for the four-element array on the cylindrical-shaped surface with $r = 10$ cm.

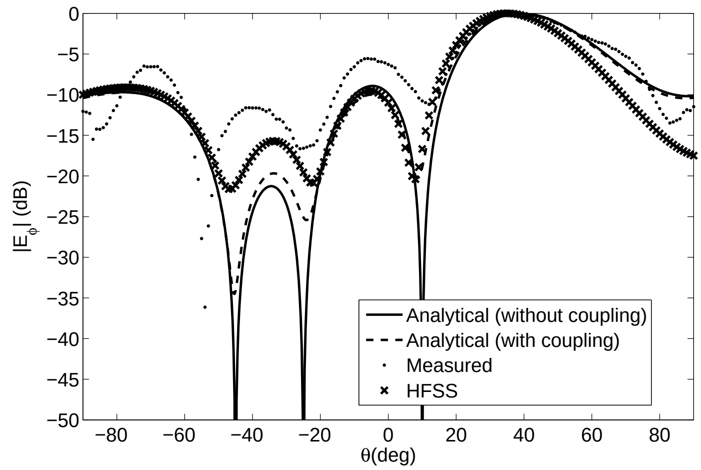

Figure 32. Pattern 2 beamforming results for the four-element array on the cylindrical-shaped surface with $r = 10$ cm.

7.5. Discussion

For the results in Figures 27-32 the most agreement is between the measured results, the HFSS simulations and the analytical computations with the array weights including mutual coupling. This illustrates the improved accuracy of the weights computed using (5.9).

Also, when comparing the pattern 2 results in Figures 28 and 30, it is shown that more disagreement between the results around $-60°$ exist for the $\theta_b = 45°$ surface than for the $\theta_b = 30°$ surface. This is thought to be be due to the more severe surface deformation that exists for the $\theta_b = 45°$ surface.

Overall though, the array weights computed using (3.23) and (5.9) have been shown to be accurate and the effects due to mutual coupling and surface deformations on the radiation pattern have been demonstrated. Then again, the weights computed using (5.9) can be used to model the coupling, and with proper optimization, an improved beamformation could be achieved in a general setting that includes the mutual coupling between elements on a changing conformal surface.

Table 2. Array weights of the conformal antenna on the wedge-shaped surface with $\theta_b = 30°$ without coupling using (4.17)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 0.9849\angle 128.76°$	$w_{-2} = 0.3113\angle -72.72°$
$w_{-1} = 0.2119\angle -75.85°$	$w_{-1} = 0.5637\angle 135.57°$
$w_1 = 1\angle 35.24°$	$w_1 = 1\angle -2.34°$
$w_2 = 0.5961\angle -174.73°$	$w_2 = 0.8440\angle -78.45°$

Table 3. Array weights of the conformal antenna on the wedge-shaped surface with $\theta_b = 30°$ with coupling using (6.9)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 0.9621\angle 88.34°$	$w_{-2} = 0.3550\angle -112.26°$
$w_{-1} = 0.2626\angle -108.76°$	$w_{-1} = 0.6054\angle 91.65°$
$w_1 = 1\angle -8.08°$	$w_1 = 1\angle -39.52°$
$w_2 = 0.6464\angle 146.62°$	$w_2 = 0.8424\angle -124.08°$

Table 4. Array weights of the conformal antenna on the wedge-shaped surface with $\theta_b = 45°$ without coupling using (4.17)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 1\angle -153.99°$	$w_{-2} = 0.2619\angle -14.85°$
$w_{-1} = 0.7462\angle -31.91°$	$w_{-1} = 0.2619\angle 165.14°$
$w_1 = 0.6266\angle 65.59°$	$w_1 = 1\angle 39.57°$
$w_2 = 0.5222\angle -107.08°$	$w_2 = 0.8801\angle -16.86°$

Table 5. Array weights of the conformal antenna on the wedge-shaped surface with $\theta_b = 45°$ with coupling using (6.9)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 1\angle{-155.04°}$	$w_{-2} = 0.3440\angle{-15.79°}$
$w_{-1} = 0.4090\angle{-56.29°}$	$w_{-1} = 0.1426\angle 121.26°$
$w_1 = 0.2436\angle 55.72°$	$w_1 = 0.1754\angle 20.97°$
$w_2 = 0.5353\angle{-116.44°}$	$w_2 = 1\angle{-28.94°}$

Table 6. Array weights of the conformal antenna on the cylindrical surface with $r = 10$ cm without coupling using (4.17)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 0.9537\angle 48.27°$	$w_{-2} = 0.5309\angle{-133.55°}$
$w_{-1} = 0.2258\angle 38.35°$	$w_{-1} = 0.8599\angle 83.89°$
$w_1 = 1\angle{-7.36°}$	$w_1 = 0.9131\angle{-51.41°}$
$w_2 = 0.6003\angle 110.01°$	$w_2 = 1\angle{-136.09°}$

Table 7. Array weights of the conformal antenna on the cylindrical surface with $r = 10$ cm with coupling using (6.9)

Pattern 1 weights	Pattern 2 weights
$w_{-2} = 0.9751\angle 5.39°$	$w_{-2} = 0.5924\angle 168.29°$
$w_{-1} = 0.2665\angle 102.15°$	$w_{-1} = 0.9428\angle 29.34°$
$w_1 = 1\angle{-75.79°}$	$w_1 = 1\angle{-102.68°}$
$w_2 = 0.9325\angle 61.92°$	$w_2 = 0.8889\angle 169.59°$

Table 8. Measured attenuation of voltage variable attenuator

Control voltage (V)	Attenuation (dB)
0.0	−34.9
0.1	−34.9
0.2	−34.9
0.3	−34.9
0.4	−34.9
0.5	−34.9
0.6	−34.9
0.7	−34.9
0.8	−34.9
0.9	−34.5
1.0	−34.1
1.1	−29.7
1.2	−27.8
1.3	−24.9
1.4	−22.4
1.5	−22.0
1.6	−19.9
1.7	−18.3
1.8	−17.7
1.9	−16.7
2.0	−16.0
2.1	−15.5
2.2	−14.9
2.3	−14.0
2.4	−13.7
2.5	−13.5
2.6	−12.8
2.7	−12.6
2.8	−12.3

Control voltage (V)	Attenuation (dB)
2.9	−11.9
3.0	−11.5
3.1	−11.2
3.2	−10.9
3.3	−10.7
3.4	−10.4
3.5	−10.3
3.6	−9.9
3.7	−9.7
3.8	−9.5
3.9	−9.3
4.0	−9.1
4.1	−8.8
4.2	−8.7
4.3	−8.6
4.4	−8.2
4.5	−8.1
4.6	−7.9
4.7	−7.7
4.8	−7.6
4.9	−7.3
5.0	−7.1
5.1	−6.9
5.2	−6.8
5.3	−6.6
5.4	−6.4
5.5	−6.3
5.6	−6.1
5.7	−5.9
5.8	−5.7
5.9	−5.6
6.0	−5.4
6.1	−5.3
6.2	−5.2
6.3	−5.1
6.4	−5.0
6.5	−5.0
6.6	−4.9
6.7	−4.9
6.8	−4.8
6.9	−4.8
7.0	−4.8

Control voltage (V)	Attenuation (dB)
7.1	−4.7
7.2	−4.7
7.3	−4.7
7.4	−4.7
7.5	−4.6
7.6	−4.6
7.7	−4.6
7.8	−4.6
7.9	−4.6
8.0	−4.5
8.1	−4.5
8.2	−4.5
8.3	−4.5
8.4	−4.5
8.5	−4.4
8.6	−4.4
8.7	−4.4
8.8	−4.4
8.9	−4.4
9.0	−4.3
9.1	−4.3
9.2	−4.3
9.3	−4.3
9.4	−4.3
9.5	−4.3
9.6	−4.3
9.7	−4.3
9.8	−4.3
9.9	−4.3
10.0	−4.2
10.1	−4.2
10.2	−4.2
10.3	−4.2
10.4	−4.2
10.5	−4.2
10.6	−4.2
10.7	−4.2
10.8	−4.2
10.9	−4.1
11.0	−4.1
11.1	−4.1
11.2	−4.1

Control voltage (V)	Attenuation (dB)
11.3	−4.1
11.4	−4.1
11.5	−4.1
11.6	−4.1
11.7	−4.1
11.8	−4.1
11.9	−4.1
12.0	−4.0
12.1	−4.0
12.2	−4.0
12.3	−4.0
12.4	−4.0
12.5	−4.0
12.6	−4.0
12.7	−4.0
12.8	−4.0
12.9	−4.0
13.0	−4.0
13.1	−4.0
13.2	−3.9
13.3	−3.9
13.4	−3.9
13.5	−3.9
13.6	−3.9
13.7	−3.9
13.8	−3.9
13.9	−3.9
14.0	−3.9
14.1	−3.9
14.2	−3.9
14.3	−3.9
14.4	−3.9
14.5	−3.9
14.6	−3.9
14.7	−3.9
14.8	−3.8
14.9	−3.8
15.0	−3.8
15.1	−3.8
15.2	−3.8
15.3	−3.8
15.4	−3.8

Control voltage (V)	Attenuation (dB)
15.5	−3.8
15.6	−3.8
15.7	−3.8
15.8	−3.8
15.9	−3.8
16.0	−3.8
16.1	−3.8
16.2	−3.8
16.3	−3.8
16.4	−3.8
16.5	−3.8
16.6	−3.8
16.7	−3.8
16.8	−3.8
16.9	−3.8
17.0	−3.8

Table 9. Measured phase shift of analog phase shifter

Control voltage (V)	Phase shift (degrees)
0.0	−82.9
0.1	−71.5
0.2	−66.4
0.3	−54.0
0.4	−48.0
0.5	−39.2
0.6	−32.8
0.7	−25.5
0.8	−16.6
0.9	−13.0
1.0	−5.3
1.1	0.0
1.2	7.0
1.3	14.2
1.4	17.3
1.5	23.0
1.6	29.6
1.7	35.2
1.8	37.6
1.9	43.7
2.0	48.9
2.1	53.5
2.2	59.3
2.3	63.1
2.4	67.7
2.5	70.6
2.6	75.8
2.7	79.7
2.8	83.8

Control voltage (V)	Phase shift (degrees)
2.9	87.0
3.0	94.8
3.1	98.7
3.2	101.8
3.3	105.9
3.4	109.0
3.5	115.4
3.6	119.1
3.7	122.8
3.8	126.8
3.9	130.5
4.0	135.6
4.1	139.7
4.2	143.0
4.3	146.7
4.4	149.6
4.5	153.2
4.6	158.0
4.7	161.2
4.8	165.4
4.9	169.6
5.0	174.0
5.1	178.3
5.2	−178.2
5.3	−176.0
5.4	−172.5
5.5	−167.0
5.6	−163.4
5.7	−161.9
5.8	−159.0
5.9	−155.9
6.0	−152.5
6.1	−148.3
6.2	−143.5
6.3	−140.8
6.4	−138.0
6.5	−133.5
6.6	−130.8
6.7	−127.3
6.8	−125.7
6.9	−123.4
7.0	−118.0

Control voltage (V)	Phase shift (degrees)
7.1	−115.7
7.2	−112.7
7.3	−110.0
7.4	−106.8
7.5	−103.2
7.6	−101.6
7.7	−98.4
7.8	−95.9
7.9	−92.2
8.0	−88.5
8.1	−86.0
8.2	−83.5
8.3	−81.9
8.4	−78.1
8.5	−77.0
8.6	−72.9
8.7	−70.8
8.8	−67.9
8.9	−64.2
9.0	−63.1
9.1	−60.7
9.2	−57.8
9.3	−55.2
9.4	−52.3
9.5	−48.7
9.6	−47.0
9.7	−44.4
9.8	−41.2
9.9	−39.4
10.0	−35.5
10.1	−33.4
10.2	−31.0
10.3	−28.3
10.4	−26.5
10.5	−23.3
10.6	−21.3
10.7	−18.6
10.8	−16.8
10.9	−15.2
11.0	−11.4
11.1	−8.9

Control voltage (V)	Phase shift (degrees)
11.2	−7.4
11.3	−5.8
11.4	−3.8
11.5	−1.9
11.6	0.0
11.7	2.0
11.8	3.1
11.9	5.0
12.0	5.8
12.1	7.9
12.2	9.2
12.3	10.3
12.4	11.1
12.5	12.6
12.6	13.6
12.7	14.2
12.8	14.7
12.9	15.8
13.0	16.4

CHAPTER 8. CONCLUSION AND FUTURE WORK

In this work, the beamforming of a conformal antenna on a changing conformal surface is investigated by relating the varying position of each antenna element in the array to the weighting coefficients (i.e, array weights) required to give a desired antenna beamformation. New matrices for computing the array weights that both do not include and include the mutual coupling between elements have been investigated. In particular, the beamforming of a 1×4 array on a changing wedge- and cylindrical-shaped surface is studied using the projection method on a wavefront of a transmitted wave in a particular direction. To validate the theory, a 1×4 prototype antenna with individual voltage-controlled phase shifters and attenuators is used to implement the computed weights of each individual antenna element for measurements in an anechoic chamber. Overall agreement between theory, simulations and measurements is shown throughout the work. Furthermore, the effects of mutual coupling and changing conformal surfaces on the behavior of the beamforming pattern and array weights is investigated and summarized.

Some research directions for future work are as follows:

- Investigate how coupling model with proper optimization could be used for an improved beamformation.
- Investigate dependence of array weights on the cconformal surface geometry for large number of antenna elements.
- Investigate beamformation patterns for other bend angles and various radius-of-curvature values.
- Extension of projection and matrix methods for 2-D adaptive beamforming.
- Investigate half-power beamwidth of beamforming pattern on wedge- and cylindrical-shaped conformal surfaces.

BIBLIOGRAPHY

[1] David M. Pozar and Daniel H. Schaubert, *Microstrip Antennas: The analysis and Design of Microstrip Antennas and Arrays,* IEEE Press, Piscataway, NJ, 1995.

[2] J. P. Daniel, "Directivity of linear microstrip arrays," *Electronics Letters,* pp. 897-899, vol. 23, no. 17, Aug. 1987.

[3] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design,* 3rd ed., John Wiley and Sons, Inc., New York, NY, 2012.

[4] R. C. Hansen, *Phased Array Antennas,* John Wiley and Sons, Inc., New York, NY, 1998.

[5] J. Ramsay, "Highlights of antenna history," *IEEE Ant. Prop. Society Newsletter,*, Vol. 6, pp. 8-20, Dec. 1981.

[6] L. Josefsson and P. Persson, *Conformal Array Antenna Theory and Design,* John Wiley and Sons, Ltd., Hoboken, New Jersey, 2006.

[7] Carl B. Dietrich, Jr., Warren L. Stutzman, Byung-Ki Kim and Kai Dietze, "Smart Antennas in Wireless Communications: Base-Station Diversity and Handset Beamforming," *IEEE Antennas and Propagation Magazine,* Vol. 42, No. 5, October 2000.

[8] Panayiotis Ioannides and C. A. Balanis, "Uniform Circular Arrays for Smart Antennas," *IEEE Antennas and Propagation Magazine,* Vol. 47, No. 4, Aug. 2005.

[9] Michael Chryssomallis, "Smart Antennas," *IEEE Antennas and Propagation Magazine,* Vol. 42, No. 3, June 2000.

[10] V. Kalinichev, "Analysis of Beam-Steering and Directive Characteristics of Adaptive Antenna Arrays for Mobile Communications," *IEEE Antennas and Propagation Magazine,* Vol. 42, No. 3, June 2001.

[11] http://www.dtic.mil/dtic/tr/fulltext/u2/744630.pdf

[12] "Special Issue on Conformal Arrays," *IEEE Transactions on Antennas and Propagation,* Vol. AP-22, No.1, Jan. 1974.

[13] E. Lier, D. Purdy and G. Kautz, US Patent 6163296, "Calibration and integrated beam control / conditioning system for phased-array antennas," Dec. 19, 2000.

[14] E. Lier, D. Purdy, J. Ashe and G. Kautz, "An On-Board Integrated Beam Conditioning System for Active Phased Array Satellite Antennas," *2000 IEEE Intern. Conf. on Phased Array Systems and Technol.,* May 21-25 2000, Dana Point, CA , pp. 509 - 512.

[15] D. Purdy, J. Ashe, E. Lier, US Patent: 2002/0171583 A1, "System and Method for Efficiently Characterizing the Elements in an Array Antenna," Nov. 21, 2002.

[16] E. Lier, M. Zemlyansky, D. Purdy and D. Farina, "Phased array calibration and characterization based on orthogonal coding: Theory and experimental validation," *2010 IEEE Intern. Sympos. on Phased Array Systems and Technol. (ARRAY),* pp. 271 - 278, Oct. 12 - 15, 2010.

[17] H. Schippers, G. Spalluto and G. Vos, "Radiation analysis of conformal phased array antennas on distorted structures," *12th International Conference on Antennas and Propagation (ICAP 2003),* Mar. 31 - Apr. 3, 2003, pp. 160-163.

[18] H. Schippers, J. Verpoorte, P. Jorna, A. Hulzinga, A. Meijerink, C. Roeloffzen, R. G. Heideman, A. Leinse and M. Wintels, "Conformal phased array with beam

forming on airborne satellite communication," *International ITG Workshop on Smart Antennas,* 26-27 Feb. 2008, pp. 343-350.

[19] H. Schippers, P. Knott, T. Deloues, P. Lacomme and M. R. Scherbarth, "Vibrating antennas and compensation techniques research in NATO/RTO/SET 087/ RTG 50", *IEEE Aerospace Conference,* Mar. 3 - 10, 2007, pp. 1-13.

[20] Davies D.E.N., "Circular arrays: Their properties and potential applications," *IEEE Proceedings of 2nd International Conference on Anetnnas and Propagation,*, pp. 1-10, April 1981.

[21] Knudsen H.L., "Bidrag till teorien for antennsystemer med hel eller delvis rotationssymmetri", Ph.D. Thesis, Teknisk Forlag, Copenhagen (in Danish), 1953.

[22] Knudsen H.L., "The field radiated by a ring quasi-array of an infinite number of tangential or radial dipoles," *Proceedings of IRE,* Vol. 41, No. 6, June 1953, pp. 781-789.

[23] Dorey J., Garnier G., and Auvray G. "RIAS, Synthetic Impulse and Antenna Radar," *International Conference on Radar,* Vol. Paris, April 24-28, 1989.

[24] "Phased Array Radars in France: Present and Future," *IEEE International Symposium on Phased Array Systems and Technology,* Oct. 15-18, 1996.

[25] *Proc. of 1st Europ. Workshop on Conformal Antennas,* Karlsruhe, Germany, 29 October 1999.

[26] *Proc. of 2nd Europ. Workshop on Conformal Antennas,* Hague, Netherlands, April 24-25, 2001.

[27] *Proc. of 3rd Europ. Workshop on Conformal Antennas,* Bonn, Germany, Oct. 22-23, 2003.

[28] *Proc. of 4th Europ. Workshop on Conformal Antennas,* Stockholm, Sweden, May. 23-24, 2005.

[29] *Proc. of 5th Europ. Workshop on Conformal Antennas,* Bristol, UK, Sept. 10-11, 2007.

[30] *Proc. of 7th Europ. Workshop on Conformal Antennas,* Zagreb, Croatia, Oct. 6-7, 2011.

[31] K. Wincza and S. Gruszczynski, "Influence of Curvature Radius on Radiation Patterns in Multibeam Conformal Antennas," *36th European Microw. Conf.,* vol., no., pp. 1410-1413, Sept. 10 - 15, 2006.

[32] K. Wincza, S. Gruszczynski and K. Sachse, "Conformal four-beam antenna arrays with reduced sidelobes," *Electronics Letters,* pp. 174-175, vol. 44, no. 3, Jan. 2008.

[33] Benjamin D. Braaten, S. Roy, Irfanullah, S. Nariyal, M. A. Aziz, Neil F. Chamberlain, Michael T. Reich and Dimitris E. Anagnostou, "A self-adapting flexible (SELFLEX) antenna array for changing conformal surface applications," *IEEE Transactions on Antennas and Propagation,* Vol. 99, No. 1, October 2012.

[34] I. Chiba, K. Hariu, S. Sato and S. Mano, "A projection method providing low sidelobe pattern in conformal array antennas," *Proc. IEEE Antennas Propag. Soc. Int. Symp. Dig.,* Jun. 26 - 30, 1989.

[35] Irfanullah, S. Nariyal, S. Roy, M. M. Masud, B. Ijaz, A. Iftikhar, , S. A. Naqvi and B. D. Braaten, "A note on the fundamental maixmum gain limit

of the projection method for conformal phased array antennas," Accepted for publication in the *IEEE Proceedings of International Conference on Wireless Information Technology and Systems (IEEE ICWITS 2012),* Maui, Hawaii, Nov. 2012.

[36] S. Nariyal, Irfanullah, S. A. Naqvi, B. Ijaz, M. M. Masud, B. Booth, K. Asirvatham and B. D. Braaten, "On the Use of Amplitude Tapering for Pattern Correction of Conformal (Curved) Antennas," Submitted for review in the *7th European Conference on Antennas and Propagation (EuCAP 2013),* Gothenburg, Sweden, Apr. 2013.

[37] R. W. P. King, *The Theory of Linear Antennas with Charts and Table for Practical Applications,* Harvard University Press, Cambridge, MA, 1956.

[38] G. A. Deschamps, "Microstrip Microwave Antennas," presented at the *3rd USAF Symposium on Antennas*, 1953.

[39] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification,* John Wiley and Sons, West Sussex, England, 2003.

[40] Rai E., Nishimoto, S., Katada, T., and Watanabe, H. "Historical Overview of Phased Array Antenna for Defence Applications in Japan," *IEEE International Symposium on Phased Array Systems and Technology,* Oct. 15-18, 1996, pp.217-221.

[41] Kanno M., Hashimura, T., Katada, T., Sato, M., Fukutani, K., and Suzuki, A., "Digital Beamforming for Conformal Active Array Antenna," *IEEE International Symposium on Phased Array Systems and Technology,* Oct. 15-18, 1996, pp.37-40.

[42] Steyskal H. "Pattern Synthesis for a Conformal Wing Array," *IEEE Aerospace Conference Proceedings,* Vol. 2, 2002, pp.819-824.

[43] Constantine A. Balanis, "Antenna Theory: Analysis and Design," John Wiley and Sons, Ltd., Hoboken, New Jersey, 2005.

[44] J. C. Liberti and T. S. Rappaport, "Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications," Prentice Hall PTR, 1999.

[45] T. S. Rappaport, *Smart Antennas: Adaptive Arrays, Algorithms, and Wireless Position Location,* New York, IEEE Press, 1998.

[46] R.E. Munson, "Conformal microstrip antennas and microstrip phased arrays," *IEEE Transactions on Antennas and Propagation,* Vol. 22, Issue 1, January 1974, pp. 74-78.

[47] T.G. Campbell, "An extremely thin omnidirectional microwave antenna array for spacecraft applications," NASA Tech. Note D-5539, Nov. 1969.

[48] R.B. Waterhouse, S.D. Targonski and D.M. Kokotoff, "Design and performance of small printed antennas," *IEEE Transactions on Antennas and Propagation,* Vol. 46, No. 11, November 1998.

[49] A. Shackelford, K.F. Lee, D. Chatterjee, Y.X. Guo, K.M. Luk and R. Chair, "Small-size wide-bandwidth microstrip patch antennas," *IEEE Antennas and Propagation Society International Symposium, 2001,* Vol. 1, July 8-13, 2001, pp. 86-89.

[50] I.J. Bahl and P. Bhartia "Microstrip Antennas," Artech House, Inc. Dedham, MA, 1980, p.214-215.

[51] William Palm III, "Introduction to MATLAB 6 for Engineers," 1^{st} Ed., McGraw-Hill, New York, NY, 2001, p. 419-505.

[52] www.agilent.com.

[53] P.L. O'Donovan and A.W. Rudge, "Adaptive control of a flexible linear array," *Electron. Lett.,* vol.9, no.6, pp.121-122, Mar. 22, 1973.

[54] D. J. Chung, S. K. Bhattacharya, G. E. Ponchak and J. Papapolymerou, "An 8x8 lightweight flexible multilayer antenna array," *Proc. IEEE Antennas Propag. Soc. Int. Symp. Dig.,* June 1-5, 2009, Charleston, SC.

[55] R. L. Haupt, *Antenna Arrays: A Computational Approach,* John Wiley and Sons, Ltd., Hoboken, New Jersey, 2010.

[56] R. A. Monzingo, R. L. Haupt and T. W. Miller, "Introduction to Adaptive Arrays," SciTech, 2011.

[57] S. Nikolaou, G. E. Ponchak, J. Papapolymerou and M. M. Tentzeris, "Conformal double exponentially tapered slot antenna (DETSA) on LCP or UWB applications," *IEEE Trans. on Antennas Propag.,* pp. 1663 - 1669, vol. 54, no. 6, Jun. 2006.

[58] J. - L. Guo and J. - Y. Li, "Pattern synthesis of conformal array antenna in the presence of platform using differential evolution algorithm," *EEE Trans. on Antennas Propag.,* pp. 2615 - 2621, vol. 57, no. 9, Sep. 2009.

[59] M. A. Aziz, S. Roy, L. A. Berge, Irfanullah and B. D. Braaten, "A conformal CPW folded slot antenna array printed on a Kapton substrate," Accepted for publication in the *2012 European Conf. on Antennas Propag. (EuCAP),* Prague, Czech. Republic, Mar. 2012.

[60] R. H. Roy, "An overview of Smart Antenna Technology: The Next Wave in Wireless Communications," *IEEE Aerospace Conference,* pp. 339-345, Vol. 3, May 1998.

[61] H. Krim and M. Viberg, "Two Decades of Array Signal Processing Research: The Parametric Approach," *IEEE Signal Process. Magazine,* pp. 67-94, July 1996.

[62] International Engineering Consortium, Smart Antenna Systems, [online tutorial] http://www.iec.org/

[63] J. - S. Roh, Y. - S. Chi, J. - H. Lee, Y. Tak, S. Nam and T. J. Kang, "Embroidered wearable multiresonant folded dipole antenna for RF reception," *IEEE Antennas Wireless Propag. Lett.,* pp. 803 - 806, vol. 9, 2010.

[64] P. Jorna, H. Schippers and J. Verpoorte, "Beam synthesis for conformal array antennas with efficient tapering," *Proc. of 5th Europ. Workshop on Conformal Antennas,* Bristol, Sept. 11-12, 2007.

[65] P. Knott, "Deformation and vibration of conformal antenna arrays and compensation techniques," *Meeting proc. RTO-MP-AVT-141,* paper 19, pp. 1-12, 2006.

[66] T. J. Seidel, W. S. T. Rowe and K. Ghorbani, "Passive compensation of beam shift in a bending array," *Prog. in Electrom. Research,* vol. 29, pp. 41-53, 2012.

[67] Rogers Corporation, [online] www.rogerscorp.com.

[68] Advanced Design System (ADS) by Agilent Technologies, [online] www.agilent.com.

[69] Federal Communications Commission, [online] www.fcc.gov.

[70] Ansys Inc., Ansoft HFSS, Version 13.0.1, [online] www.ansoft.com

[71] C. A. Balanis, *Antenna Theory: Analysis and Design,* 3rd Ed., John Wiley and Sons, Hoboken, NJ, 2005, pp. 945-995.

[72] A. J. Lockyer, K. H. Alt, D. P. Coughlin, M. D. Durham, J. N. Kudva, A. C. Goets and J. Tuss, "Design and developement of a conformal load-bearing smart skin antenna: overiew of the AFRL smart skin structure technology demonstration (S3TD)", *Proc. SPIE 3674, Smart Structures and Materials 1999: Industrial and Commercial Applications of Smart Structures Technologies,* 410 Jul. 9, 1999,

[73] S. Bellofiore, C. A. Balanis, J. Foutz and A. S. Spanias, "Smart-antenna systems for mobile communicaiton networks part 1: overview and antenna design," *IEEE Antennas and Propaga. Mag.,* vol. 44, no. 3, Apr. 15, 2003, pp. 145-154.

[74] C. A. Balanis, "Smart antennas for future reconfigurable wireless communication networks," *IEEE Topical Conf. on Wireless Comm. Tech.,* Oct. 15-17, 2003, pp. 181-182.

[75] A. J. Lockyer, K. H. Alt, R. W. Kinslow, H.-P. Kan, J. N. Kudva, J. Tuess and A. C. D. P. Coughlin, M. D. Durham, J. N. Kudva, A. C. Goets and J. Tuss, "Development of a structurally integrated conformal load-bearing multifunction antenna: overview of the Air Force Smart Skin Structures Technology Demonstration Program," *Proc. SPIE 2722, Smart Structures and Materials 1996: Smart Electronics and MEMS,* 55 May 20, 1996.

[76] V. K. Varadan and V. V. Varadan, "Design and development of conformal smart spiral antenna," *Proc. SPIE 2722, Smart Structures and Materials 1996: Smart Electronics and MEMS* 46 May 20, 1996.

[77] M. Beneditti, R. Azaro and A. Massa, "Experimental validation of full-adaptive smart antenna prototype," *Elec. Lett.,* vol. 44, no 11, May 22, 2008, pp. 661-662.

[78] T. Ohira, "Electronically steerable passive array radiator antennas for low-cost analog adaptive beamforming," *2000 IEEE Int. Conf. on Phased Array Sys. and Tech.,* May 21-25, 2000, pp. 101-104.

[79] A. J. Lockyer, K. H. Alt, R. W. Kinslow, H.-P. Kan, J. N. Kudva, J. Tuess and A. C. D. P. Coughlin, M. D. Durham, J. N. Kudva, A. C. Goets and J. Tuss, "Development of a structurally integrated conformal load-bearing multifunction antenna: overview of the Air Force Smart Skin Structures Technology Demonstration Program," *Proc. SPIE 2722, Smart Structures and Materials 1996: Smart Electronics and MEMS,* 55 May 20, 1996.

[80] S. Choi and D. Shim, "A novel adaptive beamforming algorithm for a smart antenna system in a CDMA mobile communication environment," *IEEE Trans. on Vehicular Tech.,* vol. 49, no. 5, Aug. 2002, pp. 1793-1806.

[81] H. Schippers, J. Verpoorte, P. Jorna, A. Hulzinga, A. Meijerink, C. Roeloffzen, R. G. Heideman, A. Leinse and M. Wintels, "Conformal phased array with beamforming on airborne satellite communication," *International ITG Workshop on Smart Antennas,* 26-27 Feb. 2008, pp. 343-350.

[82] H. Schippers, G. Spalluto and G. Vos, "Radiation analysis of conformal phased array antennas on distorted structures," *12th International Conference on Antennas and Propagation (ICAP 2003),* Mar. 31 - Apr. 3, 2003, pp. 160-163.

[83] H. Schippers, P. Knott, T. Deloues, P. Lacomme and M. R. Scherbarth, "Vibrating antennas and compensation techniques research in NATO/RTO/SET 087/ RTG 50", *IEEE Aerospace Conference,* Mar. 3 - 10, 2007, pp. 1-13.

[84] P. Knott, "Deformation and vibration of conformal antenna arrays and compensation techniques," *Meeting proc. RTO-MP-AVT-141,* paper 19, pp. 1-12, 2006.

[85] B. D. Braaten, M. A. Aziz, S.Roy, S. Nariyal, I. Irfanullah, N. F. Chamberlain, M. T. Reich and D. E. Anagnostou, "A Self-Adapting Flexible (SELFLEX) Antenna Array for Changing Conformal Surface Applications," *IEEE Trans. Antennas Propag.,* vol. 61, no. 2, Feb., 2013, pp. 655 - 665.

[86] P. L. O'Donovan and A.W. Rudge, "Adaptive control of a flexible linear array," *Electron. Lett.,* vol.9, no.6, pp.121-122, Mar. 22, 1973.

[87] J. - L. Guo and J. - Y. Li, "Pattern synthesis of conformal array antenna in the presence of platform using differential evolution algorithm," *EEE Trans. on Antennas Propag.,* pp. 2615 - 2621, vol. 57, no. 9, Sep. 2009.

[88] Y. Wu, K. F. Warnick and C. Jin, "Design study of an L-band phaed array feed for wide-field surveys and vibration compensation on FAST," *IEEE Trans. Antennas Propag.,* vol. 61, no. 6, Jun., 2013, pp. 3026 - 3033.

[89] D. Purdy, J. Ashe, E. Lier, US Patent: 2002/0171583 A1, "System and Method for Efficiently Characterizing the Elements in an Array Antenna," Nov. 21, 2002.

[90] P. Yang, F. Yang, Z.-P. Nie, B. Li, and X. Tang, "Robust adaptive beamformer using interpolation technique for conformal antenna array," *Progress In Electromagnetics Research B,* vol. 23, pp. 215-228, 2010.

[91] T. J. Seidel, W. S. T. Rowe and K. Ghorbani, "Passive compensation of beam shift in a bending array," *Prog. in Electrom. Research,* vol. 29, pp. 41-53, 2012.

[92] P. Salonen, Y. Rahmat-Samii, M. Schaffrath, and M. Kivikoski, "Effect of textile materials on wearable antenna performance: A case study of GPS antennas," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.,* 2004, vol. 1, pp. 459-462.

[93] I. J. Gupta and A. A. Ksienski, "Effect of mutual coupling on the performance of adaptive arrays," *IEEE Trans. on Antennas and Propaga.,* vol. 31, no. 5, Sep. 1983, pp. 785-791.

[94] Ansys Inc., Ansoft HFSS, Version 13.0.1, [online] www.ansoft.com

[95] Hittite Microwave Corporation, [online] www.hittite.com.

[96] Mini-Circuits [Online]. Available: www.minicircuits.com

APPENDIX A. ARRAY FACTOR OF THE TWO-ELEMENT ADAPTIVE ARRAY IN FIGURE 11

The following MATLAB code plots the result in Figure 12.

```
% Two-element linear adaptive beamforming Matlab code
clc
close all;
clear all
format long
f = 2.45e9;
c = 3e8;
lambda = c/f;
N=2;                        % Number of array elements
d = .5*lambda;              % Spacing between elements
k = 2*pi/lambda;
theta_SOI = 0*(pi/180);     % Desired signal
theta_SNOI = 30*(pi/180);   % Undesired signal
N1 = exp((-j*k*d/2)*sin(theta_SOI));
N2 = exp((-j*k*d/2)*sin(theta_SNOI));
P1 = exp((j*k*d/2)*sin(theta_SOI));
P2 = exp((j*k*d/2)*sin(theta_SNOI));
c1 = 1;
c2 = 0;
A = [N1,N2;P1,P2];
C = [c1;c2]';
X = C/A;
```

```
w1 = X(1);
w2 = X(2);
theta = 0:.005:pi/2;
psi1 = (-j*k*d/2).*sin(theta);
psi2 = (j*k*d/2).*sin(theta);
AF = w1.*exp(psi1) + w2.*exp(psi2);
AF_uncorr = sum(AF,1);
% AF_corrected = sum(E_corr,2);
plot(theta*180/pi,20*log10(abs(AF_uncorr)./max(abs(AF_uncorr))),'b'
     ,'linewidth',2);
%title('Linear Beamforming, SOI = 0 deg, SNOI = 30 deg');
%xlabel('\theta(degrees)')
xlabel('Observation angle (deg)')
ylabel('|AF(dB)|')
%axis([0 90 -50 0])
% hold on;
% plot(phi*180/pi,abs(AF_corrected)./max(abs(AF_corrected)),'green'
       ,'linewidth',2);
% title('AF corrected');
% xlabel('\phi(degrees)')
% ylabel('|AF corrected|')
grid on;
%%%%%%%%%%%%%%%%%%
```

APPENDIX B. FOUR-ELEMENT LINEAR ADAPTIVE BEAMFORMING

The following MATLAB code plots the result in Figure 15.

```
patch_L =[
90 0.024144799
89 0.024202698
88 0.025460208
87 0.025842421
86 0.027003252
85 0.027086285
84 0.02827785
83 0.028604263
82 0.029868525
81 0.030022325
80 0.031326955
79 0.03160479
78 0.032505428
77 0.033100665
76 0.034192131
75 0.034437357
74 0.035580972
73 0.03593466
72 0.037321
71 0.037702915
70 0.038814349
69 0.039466396
68 0.040615662
67 0.040615662
66 0.042476821
65 0.042783566
64 0.044163537
63 0.044611412
62 0.046088218
61 0.046288318
60 0.047646567
59 0.04779875
58 0.04908691
57 0.049477331
56 0.050550611
55 0.050943974
54 0.052210827
53 0.052682467
52 0.053998819
51 0.054350309
50 0.055719146
49 0.05613478
48 0.057416637
47 0.057763169
46 0.058954634
45 0.059491025
44 0.060579517
43 0.060974973
42 0.062181954
41 0.062499298
40 0.063711515
39 0.064249284
38 0.065476714
37 0.065858084
36 0.066973941
35 0.067451748
34 0.068270078
33 0.068673615
32 0.06963245
31 0.070235433
30 0.070750539
29 0.071099259
28 0.071792151
27 0.072064347
26 0.072792515
25 0.073158474
24 0.073600734
23 0.074114877
22 0.074451753
21 0.074600142
20 0.074977095
19 0.075218798
18 0.075309892
17 0.075624082
16 0.075713093
15 0.075689451
14 0.075987287
13 0.07599516
12 0.076245802
11 0.07625589
10 0.076442222
9 0.076761236
8 0.0768153
7 0.077209686
6 0.077446193
5 0.077973986
4 0.077996139
```

```
3 0.078266176
2 0.078524608
1 0.078770171
0 0.079278325
-1 0.079549623
-2 0.079675862
-3 0.079920744
-4 0.080248813
-5 0.080436976
-6 0.080878403
-7 0.081102171
-8 0.081383502
-9 0.081495407
-10 0.081558238
-11 0.081582968
-12 0.080909581
-13 0.08084502
-14 0.080991533
-15 0.080803691
-16 0.080601674
-17 0.080335719
-18 0.07993881
-19 0.080503846
-20 0.080525703
-21 0.079994477
-22 0.079575728
-23 0.07935873
-24 0.078987676
-25 0.078824983
-26 0.078505742
-27 0.078491266
-28 0.078120014
-29 0.077997325
-30 0.077385804
-31 0.077318966
-32 0.076928788
-33 0.076902185
-34 0.076584109
-35 0.076328016
-36 0.076082693
-37 0.076115779
-38 0.075605604
-39 0.075518667
-40 0.07481973
-41 0.074784142
-42 0.074027881
-43 0.073523388
-44 0.072886041
-45 0.071966474
-46 0.070879909
-47 0.06966648
-48 0.068674259
-49 0.067274731
-50 0.066246732
-51 0.064659087
-52 0.063814256
-53 0.061993099
-54 0.06113115
-55 0.059305228
-56 0.058723802
-57 0.057232522
-58 0.056433756
-59 0.055095527
-60 0.054279074
-61 0.053109335
-62 0.053059373
-63 0.051562675
-64 0.050502174
-65 0.049172733
-66 0.048598564
-67 0.047001807
-68 0.046038091
-69 0.044832706
-70 0.043953753
-71 0.042776597
-72 0.041687643
-73 0.040777002
-74 0.03963423
-75 0.038679914
-76 0.037789898
-77 0.036467342
-78 0.035774145
-79 0.034657647
-80 0.033890048
-81 0.032423642
-82 0.031534522
-83 0.030522441
-84 0.029431902
-85 0.028543507
-86 0.027518828
-87 0.026507558
-88 0.025561485
-89 0.024970803
-90 0.024341084
];
```

flat_case2_meas = {
90 0.004178841
89 0.004428091
88 0.004925039
87 0.005230579
86 0.005451382
85 0.00571653
84 0.005974008
83 0.006476511
82 0.006528059
81 0.006713558
80 0.007042773
79 0.007065189
78 0.007059268
77 0.007406107
76 0.007371889
75 0.007509358
74 0.007569094
73 0.007500692
72 0.007630044
71 0.007730213
70 0.007798932
69 0.007591736
68 0.00779716
67 0.007710387
66 0.007736525
65 0.007762983
64 0.007702506
63 0.007701099
62 0.007413842
61 0.007311047
60 0.007501069
59 0.006943238
58 0.006790176
57 0.006601613
56 0.006457391
55 0.00605359
54 0.005802025
53 0.005363554
52 0.005089504
51 0.004783626
50 0.004428172
49 0.00404494
48 0.003725093
47 0.003537821
46 0.003024429
45 0.002958658
44 0.002823405
43 0.002771293
42 0.002779514
41 0.002936561
40 0.003217482
39 0.003959221
38 0.004151317
37 0.004726628
36 0.00467064
35 0.005455118
34 0.005601592
33 0.006424173
32 0.006373272
31 0.006588426
30 0.006971204
29 0.006985498
28 0.006610779
27 0.006680981
26 0.006489406
25 0.005645648
24 0.005582324
23 0.005108217
22 0.00428802
21 0.003681107
20 0.002213897
19 0.002068933
18 0.001504575
17 0.001109036
16 0.002314823
15 0.003080714
14 0.004397075
13 0.005246818
12 0.00641813
11 0.007561151
10 0.008655401
9 0.009554146
8 0.010558906
7 0.011190835
6 0.011948117
5 0.012271322
4 0.012666064
3 0.012854754
2 0.012824353
1 0.012731188
0 0.012226641
-1 0.011719287
-2 0.011059824
-3 0.010228608
-4 0.009219035
-5 0.008281623
-6 0.007342474
-7 0.006455917
-8 0.006254719
-9 0.006594821
-10 0.007623901
-11 0.009001702
-12 0.010873614

```
-13 0.013046768
-14 0.015339251
-15 0.017570484
-16 0.020069086
-17 0.022619093
-18 0.024813856
-19 0.027323744
-20 0.029548038
-21 0.031611005
-22 0.033862544
-23 0.035837158
-24 0.037267648
-25 0.039581443
-26 0.041000533
-27 0.042501301
-28 0.043525737
-29 0.044601749
-30 0.044884835
-31 0.045509278
-32 0.046735652
-33 0.046909374
-34 0.047263319
-35 0.046458055
-36 0.046722483
-37 0.046571156
-38 0.04653094
-39 0.0460037
-40 0.045481931
-41 0.044695678
-42 0.043229388
-43 0.042854284
-44 0.040957671
-45 0.040478147
-46 0.039339298
-47 0.037833048
-48 0.036904566
-49 0.035619417
-50 0.034214551
-51 0.033011087
-52 0.031462827
-53 0.03062015
-54 0.029297466
-55 0.028272739
-56 0.027004954
-57 0.02610238
-58 0.024999088
-59 0.02404854
-60 0.023016938
-61 0.021717568
-62 0.021118947
-63 0.020017596
-64 0.019038579
-65 0.018331378
-66 0.017459813
-67 0.016519591
-68 0.015740348
-69 0.01479844
-70 0.014052392
-71 0.0132248
-72 0.012538362
-73 0.01194036
-74 0.011462586
-75 0.010692009
-76 0.01010407
-77 0.009644152
-78 0.008964489
-79 0.008487271
-80 0.007924145
-81 0.007568791
-82 0.007302031
-83 0.00677906
-84 0.006512075
-85 0.006248755
-86 0.005870863
-87 0.005554268
-88 0.005326073
-89 0.005191894
-90 0.004866533
];
```

```
flat_case2_sim = [-1.62E+01      -2.78E+01      -1.98E+01
-1.80E+02      -1.67E+02      -1.54E+02      -1.41E+02
-1.91E+01      -1.65E+01      -2.83E+01      -1.96E+01
-1.79E+02      -1.66E+02      -1.53E+02      -1.40E+02
-1.82E+01      -1.69E+01      -2.80E+01      -1.94E+01
-1.78E+02      -1.65E+02      -1.52E+02      -1.39E+02
-1.75E+01      -1.74E+01      -2.71E+01      -1.93E+01
-1.77E+02      -1.64E+02      -1.51E+02      -1.38E+02
-1.69E+01      -1.80E+01      -2.60E+01      -1.93E+01
-1.76E+02      -1.63E+02      -1.50E+02      -1.37E+02
-1.64E+01      -1.87E+01      -2.49E+01      -1.93E+01
-1.75E+02      -1.62E+02      -1.49E+02      -1.36E+02
-1.61E+01      -1.95E+01      -2.38E+01      -1.93E+01
-1.74E+02      -1.61E+02      -1.48E+02      -1.35E+02
-1.59E+01      -2.04E+01      -2.29E+01      -1.94E+01
-1.73E+02      -1.60E+02      -1.47E+02      -1.34E+02
-1.57E+01      -2.14E+01      -2.21E+01      -1.94E+01
-1.72E+02      -1.59E+02      -1.46E+02      -1.33E+02
-1.56E+01      -2.26E+01      -2.15E+01      -1.95E+01
-1.71E+02      -1.58E+02      -1.45E+02      -1.32E+02
-1.56E+01      -2.39E+01      -2.09E+01      -1.96E+01
-1.70E+02      -1.57E+02      -1.44E+02      -1.31E+02
-1.57E+01      -2.53E+01      -2.05E+01      -1.97E+01
-1.69E+02      -1.56E+02      -1.43E+02      -1.30E+02
-1.59E+01      -2.67E+01      -2.01E+01      -1.98E+01
-1.68E+02      -1.55E+02      -1.42E+02      -1.29E+02
```

-1.98E+01	-1.60E+01	-1.27E+01	-5.00E+01
-1.28E+02	-1.15E+02	-1.02E+02	-8.90E+01
-1.98E+01	-1.56E+01	-1.27E+01	-2.80E+01
-1.27E+02	-1.14E+02	-1.01E+02	-8.80E+01
-1.97E+01	-1.52E+01	-1.27E+01	-2.20E+01
-1.26E+02	-1.13E+02	-1.00E+02	-8.70E+01
-1.96E+01	-1.49E+01	-1.29E+01	-1.86E+01
-1.25E+02	-1.12E+02	-9.90E+01	-8.60E+01
-1.94E+01	-1.45E+01	-1.31E+01	-1.62E+01
-1.24E+02	-1.11E+02	-9.80E+01	-8.50E+01
-1.92E+01	-1.42E+01	-1.35E+01	-1.46E+01
-1.23E+02	-1.10E+02	-9.70E+01	-8.40E+01
-1.89E+01	-1.39E+01	-1.40E+01	-1.33E+01
-1.22E+02	-1.09E+02	-9.60E+01	-8.30E+01
-1.85E+01	-1.36E+01	-1.47E+01	-1.23E+01
-1.21E+02	-1.08E+02	-9.50E+01	-8.20E+01
-1.81E+01	-1.34E+01	-1.58E+01	-1.15E+01
-1.20E+02	-1.07E+02	-9.40E+01	-8.10E+01
-1.77E+01	-1.32E+01	-1.72E+01	-1.09E+01
-1.19E+02	-1.06E+02	-9.30E+01	-8.00E+01
-1.73E+01	-1.30E+01	-1.92E+01	-1.04E+01
-1.18E+02	-1.05E+02	-9.20E+01	-7.90E+01
-1.69E+01	-1.28E+01	-2.24E+01	-9.95E+00
-1.17E+02	-1.04E+02	-9.10E+01	-7.80E+01
-1.65E+01	-1.27E+01	-2.81E+01	-9.62E+00
-1.16E+02	-1.03E+02	-9.00E+01	-7.70E+01

-9.36E+00	-9.57E+00	-1.72E+01	-2.09E+01
-7.60E+01	-6.30E+01	-5.00E+01	-3.70E+01
-9.15E+00	-9.81E+00	-1.85E+01	-2.00E+01
-7.50E+01	-6.20E+01	-4.90E+01	-3.60E+01
-8.99E+00	-1.01E+01	-2.01E+01	-1.92E+01
-7.40E+01	-6.10E+01	-4.80E+01	-3.50E+01
-8.88E+00	-1.04E+01	-2.20E+01	-1.87E+01
-7.30E+01	-6.00E+01	-4.70E+01	-3.40E+01
-8.80E+00	-1.08E+01	-2.45E+01	-1.84E+01
-7.20E+01	-5.90E+01	-4.60E+01	-3.30E+01
-8.76E+00	-1.12E+01	-2.81E+01	-1.82E+01
-7.10E+01	-5.80E+01	-4.50E+01	-3.20E+01
-8.75E+00	-1.17E+01	-3.35E+01	-1.81E+01
-7.00E+01	-5.70E+01	-4.40E+01	-3.10E+01
-8.78E+00	-1.22E+01	-3.56E+01	-1.82E+01
-6.90E+01	-5.60E+01	-4.30E+01	-3.00E+01
-8.83E+00	-1.28E+01	-3.34E+01	-1.85E+01
-6.80E+01	-5.50E+01	-4.20E+01	-2.90E+01
-8.91E+00	-1.35E+01	-2.97E+01	-1.90E+01
-6.70E+01	-5.40E+01	-4.10E+01	-2.80E+01
-9.03E+00	-1.42E+01	-2.65E+01	-1.97E+01
-6.60E+01	-5.30E+01	-4.00E+01	-2.70E+01
-9.18E+00	-1.51E+01	-2.40E+01	-2.07E+01
-6.50E+01	-5.20E+01	-3.90E+01	-2.60E+01
-9.35E+00	-1.61E+01	-2.22E+01	-2.20E+01
-6.40E+01	-5.10E+01	-3.80E+01	-2.50E+01

-2.38E+01	-1.23E+01	-1.04E+01	-1.31E+01
-2.40E+01	-1.10E+01	2.00E+00	1.50E+01
-2.65E+01	-1.16E+01	-1.11E+01	-1.12E+01
-2.30E+01	-1.00E+01	3.00E+00	1.60E+01
-3.11E+01	-1.10E+01	-1.19E+01	-9.68E+00
-2.20E+01	-9.00E+00	4.00E+00	1.70E+01
-3.57E+01	-1.04E+01	-1.29E+01	-8.37E+00
-2.10E+01	-8.00E+00	5.00E+00	1.80E+01
-3.41E+01	-1.00E+01	-1.42E+01	-7.24E+00
-2.00E+01	-7.00E+00	6.00E+00	1.90E+01
-2.77E+01	-9.68E+00	-1.60E+01	-6.26E+00
-1.90E+01	-6.00E+00	7.00E+00	2.00E+01
-2.36E+01	-9.44E+00	-1.84E+01	-5.40E+00
-1.80E+01	-5.00E+00	8.00E+00	2.10E+01
-2.08E+01	-9.29E+00	-2.21E+01	-4.64E+00
-1.70E+01	-4.00E+00	9.00E+00	2.20E+01
-1.86E+01	-9.23E+00	-2.77E+01	-3.96E+00
-1.60E+01	-3.00E+00	1.00E+01	2.30E+01
-1.69E+01	-9.26E+00	-3.09E+01	-3.37E+00
-1.50E+01	-2.00E+00	1.10E+01	2.40E+01
-1.55E+01	-9.39E+00	-2.35E+01	-2.84E+00
-1.40E+01	-1.00E+00	1.20E+01	2.50E+01
-1.42E+01	-9.63E+00	-1.87E+01	-2.37E+00
-1.30E+01	0.00E+00	1.30E+01	2.60E+01
-1.32E+01	-9.97E+00	-1.55E+01	-1.95E+00
-1.20E+01	1.00E+00	1.40E+01	2.70E+01

-1.58E+00	-8.07E-02	-2.41E+00	-6.82E+00
2.80E+01	4.10E+01	5.40E+01	6.70E+01
-1.26E+00	-1.53E-01	-2.69E+00	-7.22E+00
2.90E+01	4.20E+01	5.50E+01	6.80E+01
-9.85E-01	-2.46E-01	-2.99E+00	-7.62E+00
3.00E+01	4.30E+01	5.60E+01	6.90E+01
-7.47E-01	-3.59E-01	-3.29E+00	-8.03E+00
3.10E+01	4.40E+01	5.70E+01	7.00E+01
-5.45E-01	-4.91E-01	-3.61E+00	-8.45E+00
3.20E+01	4.50E+01	5.80E+01	7.10E+01
-3.78E-01	-6.41E-01	-3.93E+00	-8.88E+00
3.30E+01	4.60E+01	5.90E+01	7.20E+01
-2.43E-01	-8.09E-01	-4.26E+00	-9.32E+00
3.40E+01	4.70E+01	6.00E+01	7.30E+01
-1.39E-01	-9.94E-01	-4.61E+00	-9.77E+00
3.50E+01	4.80E+01	6.10E+01	7.40E+01
-6.50E-02	-1.19E+00	-4.96E+00	-1.02E+01
3.60E+01	4.90E+01	6.20E+01	7.50E+01
-1.81E-02	-1.41E+00	-5.31E+00	-1.07E+01
3.70E+01	5.00E+01	6.30E+01	7.60E+01
2.52E-03	-1.64E+00	-5.68E+00	-1.12E+01
3.80E+01	5.10E+01	6.40E+01	7.70E+01
-1.77E-03	-1.88E+00	-6.06E+00	-1.18E+01
3.90E+01	5.20E+01	6.50E+01	7.80E+01
-2.99E-02	-2.14E+00	-6.44E+00	-1.23E+01
4.00E+01	5.30E+01	6.60E+01	7.90E+01

-1.29E+01	-2.65E+01	-1.51E+01	-1.42E+01
8.00E+01	9.30E+01	1.06E+02	1.19E+02
-1.36E+01	-2.33E+01	-1.49E+01	-1.42E+01
8.10E+01	9.40E+01	1.07E+02	1.20E+02
-1.44E+01	-2.12E+01	-1.48E+01	-1.42E+01
8.20E+01	9.50E+01	1.08E+02	1.21E+02
-1.52E+01	-1.97E+01	-1.47E+01	-1.41E+01
8.30E+01	9.60E+01	1.09E+02	1.22E+02
-1.62E+01	-1.86E+01	-1.46E+01	-1.41E+01
8.40E+01	9.70E+01	1.10E+02	1.23E+02
-1.74E+01	-1.78E+01	-1.46E+01	-1.40E+01
8.50E+01	9.80E+01	1.11E+02	1.24E+02
-1.88E+01	-1.71E+01	-1.45E+01	-1.40E+01
8.60E+01	9.90E+01	1.12E+02	1.25E+02
-2.07E+01	-1.66E+01	-1.45E+01	-1.39E+01
8.70E+01	1.00E+02	1.13E+02	1.26E+02
-2.31E+01	-1.62E+01	-1.44E+01	-1.38E+01
8.80E+01	1.01E+02	1.14E+02	1.27E+02
-2.66E+01	-1.59E+01	-1.44E+01	-1.37E+01
8.90E+01	1.02E+02	1.15E+02	1.28E+02
-3.27E+01	-1.56E+01	-1.43E+01	-1.36E+01
9.00E+01	1.03E+02	1.16E+02	1.29E+02
-5.00E+01	-1.54E+01	-1.43E+01	-1.35E+01
9.10E+01	1.04E+02	1.17E+02	1.30E+02
-3.22E+01	-1.52E+01	-1.43E+01	-1.33E+01
9.20E+01	1.05E+02	1.18E+02	1.31E+02

```
-1.32E+01    1.44E+02    -1.11E+01    1.69E+02
1.32E+02     -1.09E+01   1.57E+02     -2.16E+01
-1.30E+01    1.45E+02    -1.14E+01    1.70E+02
1.33E+02     -1.07E+01   1.58E+02     -2.41E+01
-1.29E+01    1.46E+02    -1.17E+01    1.71E+02
1.34E+02     -1.06E+01   1.59E+02     -2.74E+01
-1.27E+01    1.47E+02    -1.21E+01    1.72E+02
1.35E+02     -1.05E+01   1.60E+02     -3.13E+01
-1.25E+01    1.48E+02    -1.25E+01    1.73E+02
1.36E+02     -1.05E+01   1.61E+02     -3.47E+01
-1.23E+01    1.49E+02    -1.30E+01    1.74E+02
1.37E+02     -1.05E+01   1.62E+02     -3.48E+01
-1.21E+01    1.50E+02    -1.36E+01    1.75E+02
1.38E+02     -1.04E+01   1.63E+02     -3.00E+01
-1.19E+01    1.51E+02    -1.43E+01    1.76E+02
1.39E+02     -1.05E+01   1.64E+02     -2.60E+01
-1.17E+01    1.52E+02    -1.51E+01    1.77E+02
1.40E+02     -1.05E+01   1.65E+02     -2.35E+01
-1.15E+01    1.53E+02    -1.59E+01    1.78E+02
1.41E+02     -1.06E+01   1.66E+02     -2.16E+01
-1.13E+01    1.54E+02    -1.70E+01    1.79E+02
1.42E+02     -1.08E+01   1.67E+02     -2.02E+01
-1.12E+01    1.55E+02    -1.82E+01    1.80E+02
1.43E+02     -1.09E+01   1.68E+02     -1.91E+01
-1.10E+01    1.56E+02    -1.97E+01    ];

% patch_L = [ ] is the single patch pattern at 2.47 GHz from
```

```
% measurement.
% First column represents observation angles
% (from -90 to 90 degrees)
% and second column is the actual measurement data
% flat_case2_meas = [ ] is the measurement data of the
% four element linear array for case 2
% flat_case2_sim = [ ] is the simulation (ADS) data of the
% four element linear array for case 2
% Scanning of four element linear array using equations
close all;
clear all;
clc;
format long
N =4;
c = 3e8;
f = 2.47e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = 0*pi/180;
theta_s = 40*(pi/180); % Angle from x-axis = 90 deg or angle
% from z-axis = 0 deg
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);  % Desired angle
th1 = -45*(pi/180); %-20*(pi/180); % first interferer angle
```

```
th2 = -25*(pi/180); %10*(pi/180);  % second interferer angle
th3 = 10*(pi/180); %30*(pi/180);   % third interferer angle
th_array = [th0 th1 th2 th3];

for m = 1:length(th_array)
    if th_array(m)<0
        th_pl(m) = theta_b-abs(th_array(m));
        th_pr(m) = -theta_b-abs(th_array(m));
    elseif th_array(m)>=0
        th_pl(m) = theta_b+abs(th_array(m));
        th_pr(m) = -theta_b+abs(th_array(m));
    else
        %do nothing
    end
end
 patch_L_norm = patch_L(:,2)./max(patch_L(:,2));
 patch_L(:,2) = patch_L_norm;           % Second column
 patch_L(:,1) = flipud(patch_L(:,1));   % First column

%  th_array = [th0 th1 th2 th3]*180/pi;
 th_pl = th_pl*180/pi                  % 40, -45, -25, 10
 th_pr = th_pr*180/pi                  % 40, -45, -25, 10

for n = 1:length(patch_L);
    if th_pl(1) == patch_L(n,1)
        el1 = patch_L(n,2);
```

```
        el2 = el1;
    else
        %do nothing
    end
end
for n = 1:length(patch_L);
    if th_pr(1) == patch_L(n,1)
        er3 = patch_L(n,2);
        er4 = er3;
    else
        %do nothing
    end
end

% Element pattern value at SOI = 40 deg
% el1 =  0.971750929679342;
% el2 =  0.971750929679342;
% er3 =  0.971750929679342;
% er4 =  0.971750929679342;
% Left element #1 on a wedge
psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + th0))*el1;
% Left element #2 on a wedge
psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b + th0))*el2;
% Right element #3 on a wedge
psi3_SOI = exp(-1j*k*(d/2)*sin(theta_b - th0))*er3;
% Right element #4 on a wedge
```

```
psi4_SOI = exp(-1j*k*(3*d/2)*sin(theta_b - th0))*er4;
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0

 for n = 1:length(patch_L);
    if th_pl(2) == patch_L(n,1)
        el1 = patch_L(n,2);
        el2 = el1;
    else
        %do nothing
    end
end
for n = 1:length(patch_L);
    if th_pr(2) == patch_L(n,1)
        er3 = patch_L(n,2);
        er4 = er3;
    else
        %do nothing
    end
end

% Element pattern value at SNOI = -45 deg
% el1 = 0.919028773260615;
% el2 = 0.919028773260615;
% er3 = 0.919028773260615;
% er4 = 0.919028773260615;
psi1_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + th1))*el1;
```

```
psi2_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + th1))*el2;
psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b - th1))*er3;
psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b - th1))*er4;
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1

for n = 1:length(patch_L);
    if th_pl(3) == patch_L(n,1)
        el1 = patch_L(n,2);
        el2 = el1;
    else
        %do nothing
    end
end
for n = 1:length(patch_L);
    if th_pr(3) == patch_L(n,1)
        er3 = patch_L(n,2);
        er4 = er3;
    else
        %do nothing
    end
end
% Element pattern value at SNOI = -45 deg
% el1 =  0.999696873004179;
% el2 =  0.999696873004179;
% er3 =  0.999696873004179;
% er4 =  0.999696873004179;
```

```
psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + th2))*el1;
psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + th2))*el2;
psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - th2))*er3;
psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - th2))*er4;
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2

tol = 0;                            % Not required for this case
for n = 1:length(patch_L);
    if th_pl(4)+tol == patch_L(n,1)
        el1 = patch_L(n,2);
        el2 = el1;
    else
        %do nothing
    end
end
for n = 1:length(patch_L);
    if th_pr(4)+tol == patch_L(n,1)
        er3 = patch_L(n,2);
        er4 = er3;
    else
        %do nothing
    end
end

% Element pattern value at SNOI = 10 deg
% el1 =  0.948553428455802;
```

```
% el2 =  0.948553428455802;
% er3 =  0.948553428455802;
% er4 =  0.948553428455802;
psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*el1;
psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*el2;
psi3_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b - th3))*er3;
psi4_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b - th3))*er4;
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3


% Left element #1 on a wedge
% psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + th0))*
                cos(th0 + theta_b);
% Left element #2 on a wedge
% psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b + th0))*
                cos(th0 + theta_b);
% Right element #3 on a wedge
% psi3_SOI = exp(-1j*k*(d/2)*sin(theta_b - th0))*
               cos(th0 - theta_b);
 % Right element #4 on a wedge
% psi4_SOI = exp(-1j*k*(3*d/2)*sin(theta_b - th0))*
               cos(th0 - theta_b);
% SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0
%
% psi1_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + th1))*
                cos(th1 + theta_b);
```

```
% psi2_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + th1))*cos(th1 + theta_b);
% psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b - th1))*
               cos(th1 - theta_b);
% psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b - th1))*
               cos(th1 - theta_b);
% SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
%
% psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + th2))*
              cos(th2 + theta_b);
% psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + th2))*
              cos(th2 + theta_b);
% psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - th2))*
              cos(th2 - theta_b);
% psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - th2))*
              cos(th2 - theta_b);
% SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
%
% psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*
               cos(th3 + theta_b);
% psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*
               cos(th3 + theta_b);
% psi3_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b - th3))*
               cos(th3 - theta_b);
% psi4_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b - th3))*
               cos(th3 - theta_b);
% SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3
```

```
A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
%0w1 = u*inv(A)
w = u/A;
w_mag_norm = abs(w)./max(abs(w));
w_mag_dB = 20*log10(w_mag_norm)
w_phase_deg = angle(w).*(180/pi)
% Array factor
theta = (pi/180)*(-90:1:90);
 e_r = cos(theta - theta_b);  % Element pattern
                              % for right 2 elements at
                              % theta = 45 deg from z-axis
 e_l = cos(theta + theta_b);  % Element pattern
                              % for left 2 elements at
                              % theta = -45 deg from z-axis
 u1 = sin(theta);
 u2 = cos(theta);
 x_n = [-3*d/2*cos(theta_b),-d/2*cos(theta_b),d/2*
        cos(theta_b),3*d/2*
        cos(theta_b)];
 z_n = -[3*d/2*sin(theta_b),d/2*sin(theta_b),d/2*
        sin(theta_b),3*d/2*
        sin(theta_b)];

     AP = w(1).*(patch_L_norm.').*exp(j.*k.*(x_n(1).*u1 +
```

```
          z_n(1).*u2))
          + ...
           w(2).*(patch_L_norm.').*exp(j.*k.*(x_n(2).*u1 +
          z_n(2).*u2))
          + ...
           w(3).*(patch_L_norm.').*exp(j.*k.*(x_n(3).*u1 +
           z_n(3).*u2))
          + ...
           w(4).*(patch_L_norm.').*exp(j.*k.*(x_n(4).*u1 +
           z_n(4).*u2));



AP_norm = abs(AP)./max(abs(AP));
flat_case2_meas_norm = flat_case2_meas(:,2)./
max(abs(flat_case2_meas(:,2)));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(-flat_case2_meas(:,1),20*log10(flat_case2_meas_norm),'k.');
hold on
plot(flat_case2_sim(:,1),flat_case2_sim(:,2),'kx');
```

```
xlabel('\theta (deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)','Measured','ADS')

% th = -90:1:90;
% figure
% plot(th,20*log10(patch_L_norm))
```

APPENDIX C. COMPLEX WEIGHTS FOR THE THREE-ELEMENT ARRAY ON THE WEDGE-SHAPED CONFORMAL SURFACE

The following MATLAB code plots the result in Figures 20-21.

```
% Effect of bend angles --- three elements antenna
close all;
clear all;
clc;
format short
N =3;
c = 3e8;
f = 2.47e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = (0:60)*pi/180;
theta_s = 40*(pi/180);     % SOI
th0 = theta_s; %40*(pi/180); %receive angle
th1 = -45*(pi/180); %-45*(pi/180); % first interferer angle
th2 = 10*(pi/180); %10*(pi/180);   % second interferer angle
th01 = 40;
th11 = -45;
th22 = 10;
theta_b1 = 0:60;
%############################################################
```

```
% For SOI, SNOI = or > 0
for ii = 1:length(theta_b1)
if (th01 <= theta_b1(ii))
psi1_SOI(ii) = exp(-1j.*k.*(d).*sin(theta_b(ii) + th0))
                  .*cos(th0 + theta_b(ii));
psi2_SOI(ii) = exp(-1j.*k*(0).*sin(theta_b(ii) + th0)).*cos(th0);
psi3_SOI(ii) = exp(-1j.*k*(d).*sin(theta_b(ii) - th0))
                  .*cos(th0 - theta_b(ii));
SOI1(:,ii) = [psi1_SOI(ii) psi2_SOI(ii) psi3_SOI(ii)].';
elseif (th01 > theta_b1(ii))
psi11_SOI(ii) = exp(-1j.*k.*(d).*sin(theta_b(ii) + th0))
                  .*cos(th0 + theta_b(ii));
psi22_SOI(ii) = exp(-1j.*k.*(0).*sin(theta_b(ii) + th0)).*cos(th0);
psi33_SOI(ii) = exp(1j.*k.*(d).*sin(th0 - theta_b(ii)))
                  .*cos(th0 - theta_b(ii));
SOI2(:,ii) = [psi11_SOI(ii) psi22_SOI(ii) psi33_SOI(ii)].';
else
end
end
SOI = [SOI2 SOI1(:,41:61,:)];      % a0
%############################################################
% For SOI, SNOI < 0
for ii = 1:length(theta_b1)
if (abs(th11) <= theta_b1(ii))
psi1_SNOI1(ii) = exp(-1j.*k.*(d)*sin(theta_b(ii) - abs(th1)))
                 .*cos(-abs(th1) + theta_b(ii));
```

```
psi2_SNOI1(ii) = exp(-1j.*k*(0).*sin(theta_b(ii) + abs(th1)))
                 .*cos(th1);
psi3_SNOI1(ii) = exp(-1j.*k*(d).*sin(theta_b(ii) + abs(th1)))
                 .*cos(-abs(th1) - theta_b(ii));
SNOI1_1(:,ii) = [psi1_SNOI1(ii) psi2_SNOI1(ii) psi3_SNOI1(ii)].';
elseif (abs(th11) > theta_b1(ii))
psi11_SNOI1(ii) = exp(1j.*k.*(d).*sin(-theta_b(ii) + abs(th1)))
                 .*cos(-abs(th1) + theta_b(ii));
psi22_SNOI1(ii) = exp(-1j.*k.*(0).*sin(theta_b(ii) + abs(th1)))
                 .*cos(th1);
psi33_SNOI1(ii) = exp(-1j.*k.*(d).*sin(abs(th1) + theta_b(ii)))
                 .*cos(-abs(th1) - theta_b(ii));
SNOI1_2(:,ii) = [psi11_SNOI1(ii) psi22_SNOI1(ii) psi33_SNOI1(ii)].';
else
end
end
SNOI1 = [SNOI1_2 SNOI1_1(:,46:61,:)];  % a1
%##########################################################
% For SOI, SNOI = or > 0
for ii = 1:length(theta_b1)
if (th22 <= theta_b1(ii))
psi1_SNOI2(ii) = exp(-1j.*k.*(d)*sin(theta_b(ii) + th2))
                     .*cos(th2 + theta_b(ii));
psi2_SNOI2(ii) = exp(-1j.*k*(0).*sin(theta_b(ii) + th2)).*cos(th2);
psi3_SNOI2(ii) = exp(-1j.*k*(d).*sin(theta_b(ii) - th2))
                     .*cos(th2 - theta_b(ii));
```

```
SNOI2_1(:,ii) = [psi1_SNOI2(ii) psi2_SNOI2(ii) psi3_SNOI2(ii)].';
elseif (th22 > theta_b1(ii))
psi11_SNOI2(ii) = exp(-1j.*k.*(d).*sin(theta_b(ii) + th2))
                      .*cos(th2 + theta_b(ii));
psi22_SNOI2(ii) = exp(-1j.*k.*(0).*sin(theta_b(ii) + th2)).*cos(th2);
psi33_SNOI2(ii) = exp(1j.*k.*(d).*sin(th2 - theta_b(ii)))
                      .*cos(th2 - theta_b(ii));
SNOI2_2(:,ii) = [psi11_SNOI2(ii) psi22_SNOI2(ii) psi33_SNOI2(ii)].';
else
end
end
SNOI2 = [SNOI2_2 SNOI2_1(:,11:61,:)];  % a2
%#########################################################

u = [1 0 0];
for p = 1:length(theta_b1)
    A = [SOI(:,p) SNOI1(:,p) SNOI2(:,p)];
    w(:,p) = u/A; % For each bend angle the complex weights are
                  % stored in column vectors
    w_mag_norm(:,p) = abs(w(:,p))./max(abs(w(:,p)));
    w_mag_dB(:,p) = 20*log10(w_mag_norm(:,p));
    w_phase_deg(:,p) = angle(w(:,p)).*(180/pi);
end
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
```

```
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta_b1,w_mag_dB(1,:),'k-')
hold on
plot(theta_b1,w_mag_dB(2,:),'k--')
hold on
plot(theta_b1,w_mag_dB(3,:),'kx')
axis([0 60 -15 5])
xlabel('\theta_b(deg)')
ylabel('Amplitude excitation(dB)')
legend('|w_{-1}|','|w_0|','|w_1|')
% Phase in degrees
figure
plot(theta_b1,w_phase_deg(1,:),'k-')
hold on
plot(theta_b1,w_phase_deg(2,:),'k--')
hold on
plot(theta_b1,w_phase_deg(3,:),'kx')
axis([0 60 -180 180])
xlabel('\theta_b(deg)')
ylabel('Phase excitation(deg)')
legend('w^{\circ}_{-1}','w^{\circ}_0','w^{\circ}_1')
```

APPENDIX D. COMPLEX WEIGHTS FOR THE THREE-ELEMENT ARRAY ON THE CYLINDRICAL-SHAPED CONFORMAL SURFACE

The following MATLAB code plots the result in Figures 22-23.

```
% Effect of various radius of curvature --- three elements antenna
close all;
clear all;
clc;
format short
N =3;
c = 3e8;
f = 2.47e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_s = 40*(pi/180);            % SOI
th0 = theta_s; %40*(pi/180);      %receive angle
th1 = -45*(pi/180); %-45*(pi/180); % first interferer angle
th2 = 10*(pi/180); %10*(pi/180);   % second interferer angle
r = (10:200).*1/100; % Radius of curvature varies from
                       % 10cm to 200cm
for ii = 1:length(r)
theta1(ii) = d./r(ii);
theta2(ii) = theta1(ii);
phi1(ii) = pi/2 - theta1(ii);
```

```
phi2(ii) = pi/2;
phi3(ii) = phi1(ii);
x_n1(ii) = -r(ii).*cos(phi3(ii));
x_n2(ii) = 0;
x_n3(ii) = -x_n1(ii);
z_n1(ii) = r(ii).*sin(phi3(ii));
z_n2(ii) = 0;
z_n3(ii) = z_n1(ii);
h_n1(ii) = sqrt((0-x_n1(ii))^2 + (r(ii)-z_n1(ii))^2);
h_n2(ii) = 0;
h_n3(ii) = h_n1(ii);
phi_inv1(ii) = asin((r(ii)-z_n1(ii))./h_n1(ii));
phi_inv2(ii) = 0;
phi_inv3(ii) = phi_inv1(ii);
phi_inv(ii,:) = [phi_inv1(ii) phi_inv2(ii) phi_inv3(ii)];
theta_b = phi_inv; % First row contains 3 bend angles
% corresponding to r = 10cm,
% second row contains 3 bend angles corresponding to r = 11cm
% and so on. We have total 191 radii
end
%############################################################
% theta_s = 40 deg > theta_b
for ii = 1:length(r)
if (th0 <= theta_b(ii,1))
psi1_SOI(ii) = exp(-1j.*k.*h_n1(ii).*sin(theta_b(ii,1) +
               abs(th0))).*cos(pi/2-phi1(ii)+th0);
```

```
psi2_SOI(ii) = exp(1j.*k.*h_n2(ii).*sin(theta_b(ii,2) +
abs(th0))).*cos(th0);
psi3_SOI(ii) = exp(-1j.*k.*h_n3(ii).*sin(theta_b(ii,3) -
abs(th0))).*cos(pi/2-phi3(ii)-th0);
SOI1(:,ii) = [psi1_SOI(ii) psi2_SOI(ii) psi3_SOI(ii)].';
elseif (th0 > theta_b(ii,1))
psi11_SOI(ii) = exp(-1j.*k.*h_n1(ii).*sin(theta_b(ii,1) +
abs(th0))).*cos(pi/2-phi1(ii)+th0);
psi22_SOI(ii) = exp(1j.*k.*h_n2(ii).*sin(theta_b(ii,2) +
abs(th0))).*cos(th0);
psi33_SOI(ii) = exp(1j.*k.*h_n3(ii).*sin(-theta_b(ii,3) +
abs(th0))).*cos(pi/2-phi3(ii)-th0);
SOI2(:,ii) = [psi11_SOI(ii) psi22_SOI(ii) psi33_SOI(ii)].';
else
end
end
SOI = SOI2;      % a0
% %########################################################
% For |SNOI1| =  45 > theta_b
for ii = 1:length(r)
if (abs(th1) <= theta_b(ii,1))
psi1_SNOI1(ii) = exp(1j.*k.*h_n1(ii).*sin(theta_b(ii,1) -
abs(th1))).*cos(pi/2-phi1(ii)+th1);
psi2_SNOI1(ii) = exp(1j.*k.*h_n2(ii).*sin(theta_b(ii,2) +
abs(th1))).*cos(th1);
psi3_SNOI1(ii) = exp(-1j.*k.*h_n3(ii).*sin(theta_b(ii,3) +
```

```
abs(th1))).*cos(pi/2-phi3(ii)-th1);
SNOI1_1(:,ii) = [psi1_SNOI1(ii) psi2_SNOI1(ii) psi3_SNOI1(ii)].';
elseif (abs(th1) > theta_b(ii,1))
psi11_SNOI1(ii) = exp(1j.*k.*h_n1(ii).*sin(-theta_b(ii,1) +
abs(th1))).*cos(pi/2-phi1(ii)+th1);
psi22_SNOI1(ii) = exp(1j.*k.*h_n2(ii).*sin(theta_b(ii,2) +
abs(th1))).*cos(th1);
psi33_SNOI1(ii) = exp(-1j.*k.*h_n3(ii).*sin(theta_b(ii,3) +
abs(th1))).*cos(pi/2-phi3(ii)-th1);
SNOI1_2(:,ii) = [psi11_SNOI1(ii) psi22_SNOI1(ii) psi33_SNOI1(ii)].';
else
end
end
SNOI1 = SNOI1_2;   % a1
% %#####################################################
% For SNOI2 = 10 < theta_b for r = 10 cm to 17 cm (8 rows)
for ii = 1:length(r)
    if (th2 <= theta_b(ii,1))
psi1_SNOI2(ii) = exp(-1j.*k.*h_n1(ii).*sin(theta_b(ii,1) +
abs(th2))).*cos(pi/2-phi1(ii)+th2);
psi2_SNOI2(ii) = exp(1j*k*h_n2(ii)*sin(theta_b(ii,2) +
abs(th2)))*cos(th2);
psi3_SNOI2(ii) = exp(-1j.*k.*h_n3(ii).*sin(theta_b(ii,3) -
abs(th2))).*cos(pi/2-phi3(ii)-th2);
SNOI2_1(:,ii) = [psi1_SNOI2(ii) psi2_SNOI2(ii) psi3_SNOI2(ii)].';
    elseif (th2 > theta_b(ii,1))
```

```
psi11_SNOI2(ii) = exp(-1j.*k.*h_n1(ii).*sin(theta_b(ii,1) +
abs(th2))).*cos(pi/2-phi1(ii)+th2);
psi22_SNOI2(ii) = exp(1j.*k.*h_n2(ii).*sin(theta_b(ii,2) +
abs(th2))).*cos(th2);
psi33_SNOI2(ii) = exp(1j.*k.*h_n3(ii).*sin(-theta_b(ii,3) +
abs(th2))).*cos(pi/2-phi3(ii)-th2);
SNOI2_2(:,ii) = [psi11_SNOI2(ii) psi22_SNOI2(ii) psi33_SNOI2(ii)].';
else
end
end
SNOI2 = [SNOI2_1 SNOI2_2(:,9:191,:)];   % a2
% %#####################################################
 u = [1 0 0];
 r = 10:200;
for p = 1:length(r)
    A = [SOI(:,p) SNOI1(:,p) SNOI2(:,p)];
    w(:,p) = u/A; % For each radius of curvature the
    % complex weights are stored in column vectors
    w_mag_norm(:,p) = abs(w(:,p))./max(abs(w(:,p)));
    w_mag_dB(:,p) = 20*log10(w_mag_norm(:,p));
    w_phase_deg(:,p) = angle(w(:,p)).*(180/pi);
end
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
```

```
    'DefaultLineLineWidth',2);
plot(r,w_mag_dB(1,:),'k-')
hold on
plot(r,w_mag_dB(2,:),'k--')
hold on
plot(r,w_mag_dB(3,:),'kx')
axis([10 200 -15 5])
xlabel('r(cm)')
ylabel('Amplitude excitation(dB)')
legend('|w_{-1}|','|w_0|','|w_1|')
% Phase in degrees
 figure
plot(r,w_phase_deg(1,:),'k-')
hold on
 plot(r,w_phase_deg(2,:),'k--')
hold on
plot(r,w_phase_deg(3,:),'kx')
axis([10 200 -180 180])
xlabel('r(cm)')
ylabel('Phase excitation(deg)')
legend('w^{\circ}_{-1}','w^{\circ}_0','w^{\circ}_1')
```

APPENDIX E. PATTERN 1 BEAMFORMING RESULTS WITH $\theta_B = 30°$

The following MATLAB code plots the result in Figure 27 for the four-element array on the wedge-shaped surface with $\theta_b = 30°$.

```
corrected_case1_mean = [
90 0.006692691
88 0.007080323
86 0.007819778
84 0.008373371
82 0.009104818
80 0.009853097
78 0.010604021
76 0.010993699
74 0.011549337
72 0.012155138
70 0.0125841
68 0.013142217
66 0.013118707
64 0.013832338
62 0.014019421
60 0.014615701
58 0.014928607
56 0.015292193
54 0.015665211
52 0.015705626
50 0.015956246
48 0.015674528
46 0.015141212
44 0.014736793
42 0.013725229
40 0.012895978
38 0.011460324
36 0.009798744
34 0.008076701
32 0.005865916
30 0.003606596
28 0.0012849
26 0.002153772
24 0.004649697
22 0.007612913
20 0.010448968
18 0.013497526
16 0.016198968
14 0.01882764
12 0.021035553
10 0.023046018
8 0.024576311
6 0.025652349
4 0.026297819
2 0.026375809
0 0.026314212
-2 0.025519039
-4 0.024029896
-6 0.022591967
-8 0.020528933
-10 0.018331675
-12 0.015953291
-14 0.013460963
-16 0.010944254
-18 0.008517558
-20 0.006436512
-22 0.004832949
-24 0.003640987
-26 0.003410148
-28 0.003863726
-30 0.004389201
-32 0.005051456
-34 0.00555894
-36 0.005688009
-38 0.005711569
-40 0.005252876
-42 0.004897707
-44 0.002845428
-46 0.002675644
-48 0.002688795
-50 0.004725773
-52 0.005325033
-54 0.005720824
-56 0.006349963
-58 0.006861805
-60 0.00737798
-62 0.007734741
-64 0.00799993
-66 0.008181294
-68 0.008169831
-70 0.008269471
-72 0.00825517
```

```
-74 0.008476529
-76 0.00842451
-78 0.0086779
-80 0.008682349
-82 0.008603541
-84 0.008568125
-86 0.008327717
-88 0.008308542
-90 0.008006422
];

HFSS_case1 = [
-90 -5.909279832
-89 -5.844976835
-88 -5.785821183
-87 -5.732479173
-86 -5.685597106
-85 -5.645805565
-84 -5.613724498
-83 -5.589969
-82 -5.57515565
-81 -5.569909304
-80 -5.574870253
-79 -5.590701651
-78 -5.61809715
-77 -5.657788688
-76 -5.710554411
-75 -5.777226709
-74 -5.85870042
-73 -5.955941271
-72 -6.069994684
-71 -6.201995141
-70 -6.353176359
-69 -6.524882637
-68 -6.718581784
-67 -6.935880202
-66 -7.178540777
-65 -7.448504405
-64 -7.747916142
-63 -8.079157143
-62 -8.444883767
-61 -8.84807541
-60 -9.292092708
-59 -9.780747634
-58 -10.31838626
-57 -10.90998296
-56 -11.56123979
-55 -12.27867358
-54 -13.06964778
-53 -13.94224983
-52 -14.90478953
-51 -15.96441611
-50 -17.12374973
-49 -18.37322005
-48 -19.67498335
-47 -20.93449463
-46 -21.97083271
-45 -22.54522392
-44 -22.51455999
-43 -21.97126199
-42 -21.15239973
-41 -20.26499487
-40 -19.42797638
-39 -18.69519252
-38 -18.08662914
-37 -17.60758505
-36 -17.25817194
-35 -17.0376506
-34 -16.94631525
-33 -16.98619946
-32 -17.16104486
-31 -17.47538711
-30 -17.93187499
-29 -18.52454103
-28 -19.22308332
-27 -19.94042093
-26 -20.48549253
-25 -20.56318086
-24 -19.94866536
-23 -18.71730073
-22 -17.1578549
-21 -15.52263885
-20 -13.9456037
-19 -12.47856373
-18 -11.13378143
-17 -9.907570532
-16 -8.790885851
-15 -7.773651145
-14 -6.846384332
-13 -6.000707742
-12 -5.22941667
-11 -4.526386833
-10 -3.886435799
-9 -3.305183688
-8 -2.77892923
-7 -2.304545201
-6 -1.879392538
-5 -1.501250903
-4 -1.168263146
-3 -0.878891369
-2 -0.631882659
-1 -0.426242966
```

```
0 -0.261217976
1 -0.136280144
2 -0.051121388
3 -0.005651171
4 0
5 -0.034528647
6 -0.109843691
7 -0.226820386
8 -0.386634319
9 -0.590803964
10 -0.841247113
11 -1.140355372
12 -1.491092673
13 -1.897126315
14 -2.363002821
15 -2.894386685
16 -3.498388752
17 -4.184024428
18 -4.962862204
19 -5.84995253
20 -6.865164015
21 -8.03507315
22 -9.395423741
23 -10.99327979
24 -12.88348287
25 -15.09247025
26 -17.43281007
27 -18.95736252
28 -18.32963149
29 -16.26530377
30 -14.08341923
31 -12.19112155
32 -10.60747615
33 -9.279985781
34 -8.15668442
35 -7.196854718
36 -6.36994717
37 -5.653025173
38 -5.028630059
39 -4.483240595
40 -4.006198759
41 -3.588961749
42 -3.224575823
43 -2.907300827
44 -2.632338025
45 -2.395629653
46 -2.193708998
47 -2.023586534
48 -1.882662134
49 -1.768656366
50 -1.679555909
51 -1.613569528
52 -1.569092049
53 -1.54467447
54 -1.538998848
55 -1.550856999
56 -1.579132279
57 -1.62278397
58 -1.680833889
59 -1.752354982
60 -1.836461716
61 -1.932302126
62 -2.039051422
63 -2.155907038
64 -2.282085033
65 -2.416817722
66 -2.55935241
67 -2.708951074
68 -2.864890827
69 -3.026464988
70 -3.192984552
71 -3.363779882
72 -3.538202425
73 -3.715626293
74 -3.895449568
75 -4.077095212
76 -4.260011539
77 -4.443672206
78 -4.62757575
79 -4.81124474
80 -4.994224627
81 -5.176082412
82 -5.356405278
83 -5.5347993
84 -5.710888387
85 -5.884313543
86 -6.054732562
87 -6.2218202
88 -6.385268855
89 -6.544789765
90 -6.700114678
];
```

```
%close all;
clear all;
clc;
format short
N =4;
c = 3e8;
f = 2.47e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = 30*pi/180;
theta_s = 0*(pi/180);
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);          %receive angle
th1 = -30*(pi/180); %-20*(pi/180);   % first interferer angle
th2 = 30*(pi/180); %10*(pi/180);     % second interferer angle
th3 = 40*(pi/180); %30*(pi/180);     % third interferer angle
% For SOI, SNOI =  0
psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th0)))*
              cos(theta_b - abs(th0));
psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b - abs(th0)))*
              cos(theta_b - abs(th0));
psi3_SOI = exp(-1j*k*(d/2)*sin(theta_b + abs(th0)))*
              cos(-abs(th0) - theta_b);
psi4_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th0)))*
```

```
             cos(-abs(th0) - theta_b);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0
% For SOI, SNOI < 0
psi1_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi2_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI > 0
psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th2)))*
               cos(theta_b + abs(th2));
psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + abs(th2)))*
               cos(theta_b + abs(th2));
psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - abs(th2)))*
               cos(-theta_b + abs(th2));
psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th2)))*
               cos(-theta_b + abs(th2));
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
% For SOI, SNOI  > 0
psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*
                cos(abs(th3) + theta_b);
psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*
```

```
                cos(abs(th3) + theta_b);
psi3_SNOI3 = exp(1j*k*(d/2)*sin(-theta_b + th3))*
                cos(abs(th3) - theta_b);
psi4_SNOI3 = exp(1j*k*(3*d/2)*sin(-theta_b + th3))*
                cos(abs(th3) - theta_b);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3
A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
  %#####################################################%
w = u/A;   % Weights with out including effect of
% mutual coupling
%######################################
% Z parameters from HFSS
% Self Terms
Z11 = -7.045083-37.356893i;
% Mutual Terms
Z12 = -3.174079+2.976914i;
Z13 = -0.578495+0.547187i;
Z14 = 0.363317-0.010929i;
Z21 = Z12;
Z23 = 0.684065+0.205307i;
Z24 = -0.533723+0.461377i;
Z31 = Z13;
Z32 = Z23;
Z34 = -3.008657+3.121613i;
Z41 = Z14;
```

```
Z42 = Z24;
Z43 = Z34;
Z22 = Z11;
Z33 = Z11;
Z44 = Z11;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0;
     Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I;  %Normalized coupling matrix
Ac = Zc\A;   %inv(Zc)*A;
wc = u/Ac;
% Weights with out mutual coupling
w_mag_norm = abs(w)./max(abs(w))
%w_mag_dB = 20*log10(w_mag_norm)
w_phase_deg = angle(w).*(180/pi)
% Weights including effect of mutual coupling
wc_mag_norm = abs(wc)./max(abs(wc))
%wc_mag_dB = 20*log10(wc_mag_norm)
wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
% Element pattern for right 2 elements at theta = 45 deg
% from z-axis
 e_r3 = cos(theta-theta_b);
```

```
 e_r4 = e_r3;
 % Element pattern for left 2 elements at theta = -45 deg
 % from z-axis
 e_l1 = cos(theta+theta_b);
 e_l2 = e_l1;
 u1 = sin(theta);
 u2 = cos(theta);
 x_n = [-3*d/2*cos(theta_b),-d/2*cos(theta_b),d/2*cos(theta_b),
       3*d/2*cos(theta_b)];
 z_n = -[3*d/2*sin(theta_b),d/2*sin(theta_b),d/2*sin(theta_b),
       3*d/2*sin(theta_b)];
 figure
 % No mutual ccoupling
 AP = w(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           w(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           w(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           w(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));
       % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           wc(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           wc(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           wc(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
APc_norm = abs(APc)./max(abs(APc));
corrected_norm_case1_meas = abs(corrected_case1_meas(:,2))
```

```
                              ./max((corrected_case1_meas(:,2)));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
plot(-corrected_case1_meas(:,1),20*log10(corrected_norm_case1_meas)
     ,'k.');
hold on
plot(-HFSS_case1(:,1),HFSS_case1(:,2),'kx');
xlabel('\theta (deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)','Analytical (with coupling)'
       ,'Measured','HFSS')
```

APPENDIX F. PATTERN 2 BEAMFORMING RESULTS WITH $\theta_B = 30°$

The following MATLAB code plots the result in Figure 28 for the four-element array on the wedge-shaped surface with $\theta_b = 30°$.

```
corrected_case2_mean = [
90 0.004220868
88 0.004444672
86 0.004443148
84 0.00453669
82 0.004589679
80 0.004484053
78 0.004284591
76 0.004354331
74 0.004106122
72 0.003984888
70 0.003946051
68 0.004128244
66 0.004296232
64 0.00428141
62 0.004538039
60 0.004574521
58 0.00442433
56 0.004241933
54 0.003954799
52 0.003505184
50 0.003017168
48 0.002403623
46 0.00175381
44 0.001157766
42 0.0007837
40 0.001302571
38 0.00174356
36 0.002254915
34 0.002557885
32 0.002722796
30 0.002897831
28 0.002892292
26 0.002555009
24 0.002329757
22 0.001942776
20 0.001165711
18 0.000603597
16 0.000326677
14 0.001009476
12 0.001921237
10 0.002987791
8 0.00375383
6 0.004732502
4 0.005396961
2 0.005838472
0 0.006132557
-2 0.006338963
-4 0.006017846
-6 0.005531844
-8 0.00478087
-10 0.003888337
-12 0.002861724
-14 0.002288054
-16 0.00292005
-18 0.004322997
-20 0.006318995
-22 0.008365565
-24 0.010469019
-26 0.012672781
-28 0.014748917
-30 0.016527311
-32 0.018422544
-34 0.019996418
-36 0.021548117
-38 0.022778625
-40 0.023763395
-42 0.024377632
-44 0.024791414
-46 0.025069236
-48 0.025065618
-50 0.024888825
-52 0.024469079
-54 0.023894853
-56 0.023225097
-58 0.022296355
-60 0.021420613
-62 0.020503092
-64 0.019305546
-66 0.018233178
-68 0.017502542
-70 0.016737769
-72 0.015929921
-74 0.015341722
```

```
-76 0.014746527
-78 0.014178994
-80 0.013317915
-82 0.01285356
-84 0.012207997
-86 0.011562143
-88 0.011094396
-90 0.010550721
];

HFSS_case2 = [
-90 -7.915357656
-89 -7.721438441
-88 -7.521382873
-87 -7.315639042
-86 -7.104689439
-85 -6.889042871
-84 -6.669226974
-83 -6.445781519
-82 -6.219252574
-81 -5.99018759
-80 -5.759131392
-79 -5.52662304
-78 -5.293193479
-77 -5.059363893
-76 -4.825644637
-75 -4.592534651
-74 -4.360521242
-73 -4.130080133
-72 -3.901675701
-71 -3.675761332
-70 -3.452779846
-69 -3.233163959
-68 -3.017336786
-67 -2.805712359
-66 -2.598696206
-65 -2.396685991
-64 -2.200072261
-63 -2.009239321
-62 -1.824566271
-61 -1.64642822
-60 -1.475197702
-59 -1.311246299
-58 -1.154946459
-57 -1.006673506
-56 -0.866807808
-55 -0.73573709
-54 -0.613858829
-53 -0.501582723
-52 -0.39933318
-51 -0.307551787
-50 -0.22669976
-49 -0.157260335
-48 -0.099741105
-47 -0.054676332
-46 -0.022629242
-45 -0.004194367
-44 0
-43 -0.010710838
-42 -0.03703093
-41 -0.079707043
-40 -0.139532588
-39 -0.217352273
-38 -0.314067668
-37 -0.430643892
-36 -0.568117679
-35 -0.727607122
-34 -0.910323459
-33 -1.117585328
-32 -1.350836076
-31 -1.611664779
-30 -1.901831915
-29 -2.223300802
-28 -2.578276357
-27 -2.969253115
-26 -3.399075178
-25 -3.871011533
-24 -4.388851388
-23 -4.957025617
-22 -5.580762247
-21 -6.266285964
-20 -7.021073096
-19 -7.854172317
-18 -8.776590844
-17 -9.80170801
-16 -10.9455607
-15 -12.2264976
-14 -13.66267982
-13 -15.26294927
-12 -16.99865498
-11 -18.72960526
-10 -20.08035399
-9 -20.50567551
-8 -19.87269208
-7 -18.65680291
-6 -17.33258792
-5 -16.11511223
-4 -15.06378637
-3 -14.18017194
-2 -13.44940526
-1 -12.85431185
```

```
0 -12.37960056
1 -12.01276087
2 -11.74394681
3 -11.5656117
4 -11.47213876
5 -11.45953667
6 -11.52521109
7 -11.66780602
8 -11.88710544
9 -12.18398658
10 -12.56041688
11 -13.01948645
12 -13.56546174
13 -14.20383126
14 -14.94127584
15 -15.78540708
16 -16.74391121
17 -17.82226112
18 -19.01811077
19 -20.30844866
20 -21.6232869
21 -22.80552148
22 -23.59824831
23 -23.76540391
24 -23.30920182
25 -22.46746869
26 -21.49546944
27 -20.54807133
28 -19.69363323
29 -18.95404313
30 -18.33074197
31 -17.81738181
32 -17.40532391
33 -17.0858548
34 -16.85097821
35 -16.69360253
36 -16.60747406
37 -16.58699907
38 -16.62700633
39 -16.72245965
40 -16.86810927
41 -17.05806191
42 -17.28524859
43 -17.54078293
44 -17.81324131
45 -18.08797579
46 -18.34670297
47 -18.56777512
48 -18.72762939
49 -18.80372198
50 -18.77860578
51 -18.64388046
52 -18.40226078
53 -18.06668455
54 -17.65691921
55 -17.19528632
56 -16.70309212
57 -16.19850311
58 -15.69578035
59 -15.20540926
60 -14.73466006
61 -14.28826401
62 -13.86904226
63 -13.47842505
64 -13.11685364
65 -12.78407968
66 -12.47938324
67 -12.20172972
68 -11.94988219
69 -11.72248176
70 -11.51810528
71 -11.33530707
72 -11.17264904
73 -11.02872263
74 -10.90216447
75 -10.79166731
76 -10.69598715
77 -10.61394727
78 -10.54443959
79 -10.48642383
80 -10.43892478
81 -10.40102812
82 -10.3718751
83 -10.35065649
84 -10.3366063
85 -10.32899556
86 -10.32712662
87 -10.33032837
88 -10.3379526
89 -10.3493718
90 -10.36397864
];
```

```
%####################################
%close all;
clear all;
clc;
format short
N =4;
c = 3e8;
f = 2.47e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = 30*(pi/180);
theta_s = 40*(pi/180);  % Angle from x-axis = 90 deg or angle
% from z-axis = 0 deg
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);           %receive angle
th1 = -45*(pi/180); %-20*(pi/180);   % first interferer angle
th2 = -25*(pi/180); %10*(pi/180);    % second interferer angle
th3 = 10*(pi/180); %30*(pi/180);     % third interferer angle

% For SOI, SNOI = or > 0
psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + th0))*
cos(th0 + theta_b);
psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b + th0))*
cos(th0 + theta_b);
```

```
psi3_SOI = exp(1j*k*(d/2)*sin(-theta_b + th0))*
cos(th0 - theta_b);
psi4_SOI = exp(1j*k*(3*d/2)*sin(-theta_b + th0))*
cos(th0 - theta_b);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0
% For SOI, SNOI < 0
psi1_SNOI1 = exp(1j*k*(3*d/2)*sin(-theta_b + abs(th1)))*
                cos(theta_b - abs(th1));
psi2_SNOI1 = exp(1j*k*(d/2)*sin(-theta_b + abs(th1)))*
                cos(theta_b - abs(th1));
psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI < 0
psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th2)))*
                cos(theta_b - abs(th2));
psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - abs(th2)))*
                cos(theta_b - abs(th2));
psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + abs(th2)))*
                cos(-theta_b - abs(th2));
psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th2)))*
                cos(-theta_b - abs(th2));
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
% For SOI, SNOI = or > 0
```

```
psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*
cos(th3 + theta_b);
psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*
cos(th3 + theta_b);
psi3_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b - th3))*
cos(th3 - theta_b);
psi4_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b - th3))*
cos(th3 - theta_b);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3

A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
  %######################################################%
w = u/A;   % Weights with out including effect of
% mutual coupling
%######################################
% Z parameters from HFSS
% Self Terms
Z11 = -7.045083-37.356893i;
% Mutual Terms
Z12 = -3.174079+2.976914i;
Z13 = -0.578495+0.547187i;
Z14 = 0.363317-0.010929i;
Z21 = Z12;
Z23 = 0.684065+0.205307i;
Z24 = -0.533723+0.461377i;
```

```
Z31 = Z13;
Z32 = Z23;
Z34 = -3.008657+3.121613i;
Z41 = Z14;
Z42 = Z24;
Z43 = Z34;
Z22 = Z11;
Z33 = Z11;
Z44 = Z11;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0;
     Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I;  %Normalized coupling matrix
Ac = Zc\A;   %inv(Zc)*A;
wc = u/Ac;
w_mag_norm = abs(w)./max(abs(w))
%w_mag_dB = 20*log10(w_mag_norm)
w_phase_deg = angle(w).*(180/pi)
wc_mag_norm = abs(wc)./max(abs(wc))
%wc_mag_dB = 20*log10(wc_mag_norm)
wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
%theta = (pi/180)*(-90:2:90);
```

```
% Element pattern for right 2 elements at theta = 45 deg
% from z-axis
 e_r3 = cos(theta-theta_b);
 e_r4 = e_r3;
 % Element pattern for left 2 elements at theta = -45 deg
 % from z-axis
 e_l1 = cos(theta+theta_b);
 e_l2 = e_l1;
 u1 = sin(theta);
 u2 = cos(theta);
% w = [1,1,1,1];

 x_n = [-3*d/2*cos(theta_b),-d/2*cos(theta_b),d/2*cos(theta_b),
       3*d/2*cos(theta_b)];
 z_n = -[3*d/2*sin(theta_b),d/2*sin(theta_b),d/2*sin(theta_b),
       3*d/2*sin(theta_b)];

 figure
 % No mutual ccoupling
     AP = w(1).*e_l1.*exp(1i.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           w(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           w(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           w(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));
       % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           wc(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
```

```
           wc(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           wc(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
APc_norm = abs(APc)./max(abs(APc));
corrected_norm_case2_meas = abs(corrected_case2_meas(:,2))
                              ./max((corrected_case2_meas(:,2)));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
plot(-corrected_case2_meas(:,1),20*log10(corrected_norm_case2_meas)
    ,'k.');
hold on
plot(-HFSS_case2(:,1),HFSS_case2(:,2),'kx');
```

APPENDIX G. PATTERN 1 BEAMFORMING RESULTS WITH $\theta_B = 45°$

The following MATLAB code plots the result in Figure 29 for the four-element array on the wedge-shaped surface with $\theta_b = 45°$.

```
corrected_case1_mean = [
90 0.015910452
88 0.016811439
86 0.017241097
84 0.017790437
82 0.018417578
80 0.018876269
78 0.019255906
76 0.019614756
74 0.019538757
72 0.020068021
70 0.020496742
68 0.020551686
66 0.020643515
64 0.020829283
62 0.020186192
60 0.020297114
58 0.019616228
56 0.019277051
54 0.018153763
52 0.017498869
50 0.016068692
48 0.014951028
46 0.013259581
44 0.011353368
42 0.009269644
40 0.007317156
38 0.005192306
36 0.002879459
34 0.000578943
32 0.001798237
30 0.004099249
28 0.006534775
26 0.008890837
24 0.010919162
22 0.012959365
20 0.01486889
18 0.016499958
16 0.018088668
14 0.019200106
12 0.020048644
10 0.020415133
8 0.020298281
6 0.02016646
4 0.019032016
2 0.018327811
0 0.01677469
-2 0.015566668
-4 0.013698969
-6 0.011659929
-8 0.009566591
-10 0.007325754
-12 0.005386197
-14 0.003243998
-16 0.001398905
-18 0.001019148
-20 0.0024917
-22 0.003978367
-24 0.005150232
-26 0.006024717
-28 0.006815874
-30 0.007217144
-32 0.007511271
-34 0.00748713
-36 0.007516696
-38 0.007209161
-40 0.006657587
-42 0.005641956
-44 0.00459156
-46 0.003559439
-48 0.002362186
-50 0.001364653
-52 0.000741468
-54 0.001717006
-56 0.002839978
-58 0.003568427
-60 0.004370457
-62 0.005097717
-64 0.005684456
-66 0.006331588
-68 0.007116319
-70 0.007322464
-72 0.008189359
```

```
-74 0.008813356
-76 0.009144497
-78 0.009794232
-80 0.009970523
-82 0.00990223
-84 0.010024559
-86 0.00942773
-88 0.009087614
-90 0.008701712
];

HFSS_case1 = [
-90 -6.767111408
-89 -6.847054958
-88 -6.936622134
-87 -7.036336925
-86 -7.146774334
-85 -7.268567594
-84 -7.402418251
-83 -7.549109459
-82 -7.709522781
-81 -7.884658756
-80 -8.075661458
-79 -8.283847323
-78 -8.510738563
-77 -8.758101695
-76 -9.027992033
-75 -9.322805501
-74 -9.645339943
-73 -9.998869251
-72 -10.38723538
-71 -10.81496576
-70 -11.28742733
-69 -11.81103382
-68 -12.39353159
-67 -13.04440254
-66 -13.7754446
-65 -14.60162612
-64 -15.54237066
-63 -16.62352623
-62 -17.88041463
-61 -19.36245736
-60 -21.13925375
-59 -23.30225929
-58 -25.91730188
-57 -28.66101192
-56 -29.70706437
-55 -27.80706282
-54 -25.12526494
-53 -22.80129229
-52 -20.91942716
-51 -19.39050582
-50 -18.13148271
-49 -17.08227235
-48 -16.20071638
-47 -15.45693549
-46 -14.82938882
-45 -14.30232707
-44 -13.86415979
-43 -13.50640015
-42 -13.22297758
-41 -13.00979101
-40 -12.86442669
-39 -12.78599611
-38 -12.77507079
-37 -12.83370675
-36 -12.96556558
-35 -13.17615517
-34 -13.473236
-33 -13.86747453
-32 -14.37349001
-31 -15.01156352
-30 -15.81053082
-29 -16.81294204
-28 -18.08493974
-27 -19.73706355
-26 -21.97419633
-25 -25.23968102
-24 -30.72963213
-23 -37.2902122
-22 -28.91390822
-21 -23.58128411
-20 -20.13116386
-19 -17.58994843
-18 -15.58057654
-17 -13.92229606
-16 -12.51504992
-15 -11.29775392
-14 -10.23039993
-13 -9.285364775
-12 -8.44278152
-11 -7.687892248
-10 -7.009445967
-9 -6.398683298
-8 -5.848667942
-7 -5.353832091
-6 -4.909658701
-5 -4.512454158
-4 -4.159182313
-3 -3.847341249
-2 -3.57487051
-1 -3.340080523
```

```
0 -3.141598539
1 -2.978327171
2 -2.849412727
3 -2.754221398
4 -2.692321891
5 -2.663473533
6 -2.667619178
7 -2.704882481
8 -2.775569265
9 -2.88017289
10 -3.019383567
11 -3.194101659
12 -3.405454951
13 -3.654819722
14 -3.943845043
15 -4.274478939
16 -4.64899342
17 -5.070002377
18 -5.540460588
19 -6.063621304
20 -6.642909622
21 -7.281630695
22 -7.982360718
23 -8.745739218
24 -9.568160642
25 -10.43754795
26 -11.32617469
27 -12.18029475
28 -12.91059627
29 -13.3975146
30 -13.53100527
31 -13.27433224
32 -12.69030492
33 -11.89750181
34 -11.00871879
35 -10.10220293
36 -9.2226736
37 -8.391996492
38 -7.618766319
39 -6.904548128
40 -6.247454469
41 -5.644071487
42 -5.090456116
43 -4.5826327
44 -4.116825773
45 -3.689556114
46 -3.297667507
47 -2.938319745
48 -2.608966458
49 -2.307327354
50 -2.031359626
51 -1.779230802
52 -1.549293922
53 -1.340065318
54 -1.150204912
55 -0.978498838
56 -0.82384414
57 -0.685235352
58 -0.561752724
59 -0.452551981
60 -0.356855456
61 -0.273944493
62 -0.203153032
63 -0.143862273
64 -0.095496305
65 -0.057518612
66 -0.029429307
67 -0.010762971
68 -0.001086958
69 0
70 -0.007130984
71 -0.022137768
72 -0.044705905
73 -0.0745472
74 -0.111398027
75 -0.155017376
76 -0.205184639
77 -0.261697151
78 -0.324367572
79 -0.393021174
80 -0.467493148
81 -0.54762603
82 -0.633267374
83 -0.724267757
84 -0.820479221
85 -0.921754214
86 -1.02794507
87 -1.138904043
88 -1.25448387
89 -1.374538798
90 -1.498926005
];
```

```
%close all
clear all;
clc;
format short
N =4;
c = 3e8;
f = 2.4e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = 45*(pi/180);
theta_s = 0*(pi/180);
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);         %receive angle
th1 = -30*(pi/180); %-20*(pi/180);  % first interferer angle
th2 = 30*(pi/180); %10*(pi/180);    % second interferer angle
th3 = 40*(pi/180); %30*(pi/180);    % third interferer angle
% For SOI, SNOI =  0
psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th0)))*
              cos(theta_b - abs(th0));
psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b - abs(th0)))*
              cos(theta_b - abs(th0));
psi3_SOI = exp(-1j*k*(d/2)*sin(theta_b + abs(th0)))*
              cos(-abs(th0) - theta_b);
psi4_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th0)))*
```

```
              cos(-abs(th0) - theta_b);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0
% For SOI, SNOI < 0
psi1_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi2_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI > 0
psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th2)))*
                cos(theta_b + abs(th2));
psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + abs(th2)))*
                cos(theta_b + abs(th2));
psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - abs(th2)))*
                cos(-theta_b + abs(th2));
psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th2)))*
                cos(-theta_b + abs(th2));
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
% For SOI, SNOI  > 0
psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*
                cos(abs(th3) + theta_b);
psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*
```

```
                cos(abs(th3) + theta_b);
psi3_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b - th3))*
                cos(abs(th3) - theta_b);
psi4_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b - th3))*
                cos(abs(th3) - theta_b);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3
A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
  %#####################################################%
w = u/A; % Weights with out including effect of mutual coupling
%######################################
% Z parameters from HFSS
% Self Terms
Z11 = 36.34+0.759i;
Z22 = -27.1-27.878i;
Z33 = -38.51-7.43i;
Z44 = 32.9-15.17i;
% Mutual Terms
Z12 = -3.58+2.31i;
Z13 = 0.1-0.1i;
Z14 = 0.73+0.0084i;
Z21 = Z12;
Z23 = -0.914-0.062i;
Z24 = 0.172-0.147i;
Z31 = Z13;
Z32 = Z23;
```

```
Z34 = -3.416+2.44i;
Z41 = Z14;
Z42 = 0.172-0.147i;
Z43 = Z34;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0
    ;Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I;  %Normalized coupling matrix
Ac = Zc\A;   %inv(Zc)*A;
wc = u/Ac;
w_mag_norm = abs(w)./max(abs(w))
%w_mag_dB = 20*log10(w_mag_norm)
w_phase_deg = angle(w).*(180/pi)
wc_mag_norm = abs(wc)./max(abs(wc))
%wc_mag_dB = 20*log10(wc_mag_norm)
wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
%theta = (pi/180)*(-90:2:90);
% Element pattern for right 2 elements at theta = 45 deg
% from z-axis
 e_r3 = cos(theta-theta_b);
 e_r4 = e_r3;
 % Element pattern for left 2 elements at theta = -45 deg
```

```
 % from z-axis
 e_l1 = cos(theta+theta_b);
 e_l2 = e_l1;
 u1 = sin(theta);
 u2 = cos(theta);
% w = [1,1,1,1];

 x_n = [-3*d/2*cos(theta_b),-d/2*cos(theta_b),d/2*cos(theta_b),
       3*d/2*cos(theta_b)];
 z_n = -[3*d/2*sin(theta_b),d/2*sin(theta_b),d/2*sin(theta_b),
       3*d/2*sin(theta_b)];

 figure
 % No mutual ccoupling
     AP = w(1).*e_l1.*exp(1i.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
          w(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
          w(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
          w(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));
       % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
          wc(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
          wc(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
          wc(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
APc_norm = abs(APc)./max(abs(APc));
```

```
corrected_norm_case1_meas = abs(corrected_case1_meas(:,2)
                               ./max((corrected_case1_meas(:,2)));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
plot(-corrected_case1_meas(:,1),20*log10(corrected_norm_case1_meas)
    ,'k.');
hold on
plot(-HFSS_case1(:,1),HFSS_case1(:,2),'kx');
xlabel('\theta (deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)','Analytical (with coupling)'
                   ,'Measured','HFSS')
```

APPENDIX H. PATTERN 2 BEAMFORMING RESULTS WITH $\theta_B = 45°$

The following MATLAB code plots the result in Figure 30 for the four-element array on the wedge-shaped surface with $\theta_b = 45°$.

```
corrected_case2_mean = [
90 0.004815412
88 0.005253683
86 0.00561818
84 0.005974044
82 0.006010661
80 0.006091816
78 0.005966971
76 0.005967896
74 0.005582013
72 0.005247097
70 0.004623481
68 0.004084263
66 0.003606683
64 0.0027635
62 0.002254038
60 0.001681229
58 0.001260649
56 0.000613164
54 9.86E-05
52 0.000616547
50 0.001339845
48 0.002093044
46 0.002786193
44 0.003349728
42 0.004098128
40 0.004724013
38 0.004852773
36 0.005198602
34 0.005010661
32 0.004996206
30 0.004463861
28 0.004211242
26 0.00355502
24 0.003047901
22 0.00195361
20 0.00137904
18 0.000706309
16 0.00097587
14 0.00200996
12 0.00279782
10 0.003789048
8 0.004528821
6 0.00510962
4 0.005495473
2 0.005586932
0 0.005698938
-2 0.005482761
-4 0.005115543
-6 0.00474232
-8 0.003833744
-10 0.002871193
-12 0.002307958
-14 0.002245786
-16 0.003051866
-18 0.004224677
-20 0.006156508
-22 0.007833721
-24 0.00931982
-26 0.01149471
-28 0.013153445
-30 0.014968161
-32 0.016580897
-34 0.017853163
-36 0.018905985
-38 0.020509638
-40 0.021474022
-42 0.022309126
-44 0.022808765
-46 0.023598137
-48 0.023677654
-50 0.023829366
-52 0.02396324
-54 0.023931291
-56 0.023689098
-58 0.023579108
-60 0.02269134
-62 0.022719732
-64 0.022020643
-66 0.021703812
-68 0.021298287
-70 0.020755615
-72 0.019747973
```

```
-74 0.019252185   -80 0.017875189   -86 0.015379247   ];
-76 0.019038138   -82 0.016794444   -88 0.014573826
-78 0.018484792   -84 0.016196381   -90 0.013875409

                  -69 -0.782142781  -46 -0.531415174  -23 -8.556471342
HFSS_case2 = [    -68 -0.67193693   -45 -0.652148158  -22 -9.29913335
-90 -4.417298532  -67 -0.569319331  -44 -0.787177663  -21 -10.09784427
-89 -4.201997224  -66 -0.474521994  -43 -0.937121212  -20 -10.95652269
-88 -3.989844757  -65 -0.3877653    -42 -1.102630625  -19 -11.87811927
-87 -3.780965654  -64 -0.309257322  -41 -1.284393346  -18 -12.86318027
-86 -3.575494385  -63 -0.239193961  -40 -1.483135231  -17 -13.90713029
-85 -3.373575307  -62 -0.177759982  -39 -1.699625043  -16 -14.99537455
-84 -3.17536268   -61 -0.125130994  -38 -1.934680872  -15 -16.09523404
-83 -2.981020745  -60 -0.081476361  -37 -2.189178627  -14 -17.14511733
-82 -2.790723855  -59 -0.046962973  -36 -2.464062715  -13 -18.04703927
-81 -2.60465662   -58 -0.021759739  -35 -2.760358999  -12 -18.67939485
-80 -2.423014028  -57 -0.006042623  -34 -3.079190081  -11 -18.94631719
-79 -2.246001474  -56 0             -33 -3.421792977  -10 -18.83847076
-78 -2.073834654  -55 -0.003838055  -32 -3.789539234  -9 -18.44031313
-77 -1.906739237  -54 -0.017785981  -31 -4.183957554  -8 -17.87472152
-76 -1.744950278  -53 -0.042100692  -30 -4.606758951  -7 -17.2475457
-75 -1.588711302  -52 -0.077070853  -29 -5.059864411  -6 -16.62847611
-74 -1.438273047  -51 -0.123020031  -28 -5.545434793  -5 -16.05629497
-73 -1.293891839  -50 -0.180308881  -27 -6.06590228   -4 -15.54983719
-72 -1.15582764   -49 -0.249336347  -26 -6.624001684  -3 -15.11673064
-71 -1.0243418    -48 -0.330539949  -25 -7.222798128  -2 -14.75885749
-70 -0.899694612  -47 -0.424395313  -24 -7.865703991  -1 -14.47543658
```

```
0 -14.26468244
1 -14.12467881
2 -14.05383689
3 -14.051143
4 -14.11630753
5 -14.2498783
6 -14.45335701
7 -14.72934728
8 -15.08176144
9 -15.51611992
10 -16.03999242
11 -16.66365952
12 -17.40112817
13 -18.27173846
14 -19.30280623
15 -20.53418265
16 -22.02659228
17 -23.87794116
18 -26.25733226
19 -29.47389041
20 -33.95624887
21 -37.10887089
22 -33.03359478
23 -28.90625308
24 -25.97909071
25 -23.8087319
26 -22.12144328
27 -20.76623988
28 -19.65409737
29 -18.72877047
30 -17.95264418
31 -17.2993954
32 -16.74992916
33 -16.28999976
34 -15.90875402
35 -15.59780658
36 -15.35063679
37 -15.16218758
38 -15.02859579
39 -14.94701042
40 -14.91547182
41 -14.93283444
42 -14.99872263
43 -15.11351371
44 -15.27834639
45 -15.49515638
46 -15.76674462
47 -16.09688866
48 -16.49051394
49 -16.95395179
50 -17.49532699
51 -18.12514479
52 -18.85719686
53 -19.70999789
54 -20.70915069
55 -21.89143362
56 -23.31232101
57 -25.06098027
58 -27.2934495
59 -30.31606827
60 -34.81193018
61 -41.24423854
62 -37.81802691
63 -32.26085769
64 -28.701007
65 -26.17964129
66 -24.25305172
67 -22.70852281
68 -21.42981691
69 -20.34674037
70 -19.41363858
71 -18.59909541
72 -17.88052667
73 -17.24113034
74 -16.66806331
75 -16.15129735
76 -15.68287084
77 -15.25638088
78 -14.86662665
79 -14.50935101
80 -14.18104767
81 -13.87881362
82 -13.60023366
83 -13.34328857
84 -13.10628113
85 -12.88777633
86 -12.68655299
87 -12.50156483
88 -12.33190949
89 -12.17680433
90 -12.03556789
];
```

```
%close all;
clear all;
clc;
format short
N =4;
c = 3e8;
f = 2.4e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_b = 45*(pi/180);
theta_s = 40*(pi/180);   % Angle from x-axis = 90 deg or angle
%from z-axis = 0 deg
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);        %receive angle
th1 = -45*(pi/180); %-20*(pi/180); % first interferer angle
th2 = -25*(pi/180); %10*(pi/180);  % second interferer angle
th3 = 10*(pi/180); %30*(pi/180);   % third interferer angle

% For SOI, SNOI = or > 0
psi1_SOI = exp(-1j*k*(3*d/2)*sin(theta_b + th0))*
cos(th0 + theta_b);
psi2_SOI = exp(-1j*k*(d/2)*sin(theta_b + th0))*
cos(th0 + theta_b);
psi3_SOI = exp(-1j*k*(d/2)*sin(theta_b - th0))*
```

```
cos(th0 - theta_b);
psi4_SOI = exp(-1j*k*(3*d/2)*sin(theta_b - th0))*
cos(th0 - theta_b);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].';   % a0
% For SOI, SNOI < 0
psi1_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi2_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b - abs(th1)))*
                cos(theta_b - abs(th1));
psi3_SNOI1 = exp(-1j*k*(d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
psi4_SNOI1 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th1)))*
                cos(-theta_b - abs(th1));
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI < 0
psi1_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b - abs(th2)))*
                cos(theta_b - abs(th2));
psi2_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b - abs(th2)))*
                cos(theta_b - abs(th2));
psi3_SNOI2 = exp(-1j*k*(d/2)*sin(theta_b + abs(th2)))*
                cos(-theta_b - abs(th2));
psi4_SNOI2 = exp(-1j*k*(3*d/2)*sin(theta_b + abs(th2)))*
                cos(-theta_b - abs(th2));
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
% For SOI, SNOI = or > 0
psi1_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b + th3))*
```

```
cos(th3 + theta_b);
psi2_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b + th3))*cos(th3 + theta_b);
psi3_SNOI3 = exp(-1j*k*(d/2)*sin(theta_b - th3))*
cos(th3 - theta_b);
psi4_SNOI3 = exp(-1j*k*(3*d/2)*sin(theta_b - th3))*
cos(th3 - theta_b);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3

A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
  %##########################################################%
w = u/A;  % Weights with out including effect of mutual coupling
%#######################################

%#######################################
% Z parameters from HFSS
% Self Terms
 Z11 = 36.34+0.759i;
 Z22 = -27.1-27.878i;
 Z33 = -38.51-7.43i;
 Z44 = 32.9-15.17i;
% Mutual Terms
Z12 = -3.58+2.31i;
Z13 = 0.1-0.1i;
Z14 = 0.73+0.0084i;
Z21 = Z12;
```

```
Z23 = -0.914-0.062i;
Z24 = 0.172-0.147i;
Z31 = Z13;
Z32 = Z23;
Z34 = -3.416+2.44i;
Z41 = Z14;
Z42 = 0.172-0.147i;
Z43 = Z34;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0;
    Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I;  %Normalized coupling matrix
Ac = Zc\A;   %inv(Zc)*A;
%Ac = inv(Zc)*A;
wc = u/Ac;
w_mag_norm = abs(w)./max(abs(w))
%w_mag_dB = 20*log10(w_mag_norm)
w_phase_deg = angle(w).*(180/pi)
wc_mag_norm = abs(wc)./max(abs(wc))
%wc_mag_dB = 20*log10(wc_mag_norm)
wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
%theta = (pi/180)*(-90:2:90);
```

```
% Element pattern for right 2 elements at theta = 45 deg
% from z-axis
 e_r3 = cos(theta-theta_b);
 e_r4 = e_r3;
 % Element pattern for left 2 elements at theta = -45 deg
 % from z-axis
 e_l1 = cos(theta+theta_b);
 e_l2 = e_l1;
 u1 = sin(theta);
 u2 = cos(theta);
% w = [1,1,1,1];

 x_n = [-3*d/2*cos(theta_b),-d/2*cos(theta_b),d/2*cos(theta_b),
        3*d/2*cos(theta_b)];
 z_n = -[3*d/2*sin(theta_b),d/2*sin(theta_b),d/2*sin(theta_b),
        3*d/2*sin(theta_b)];

 figure
 % No mutual ccoupling
     AP = w(1).*e_l1.*exp(1i.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           w(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           w(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           w(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));
       % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           wc(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
```

```
            wc(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
            wc(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
APc_norm = abs(APc)./max(abs(APc));
corrected_norm_case2_meas = abs(corrected_case2_meas(:,2))
                              ./max((corrected_case2_meas(:,2)));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
plot(-corrected_case2_meas(:,1),20*log10(corrected_norm_case2_meas)
     ,'k.');
hold on
plot(-HFSS_case2(:,1),HFSS_case2(:,2),'kx');
xlabel('\theta(deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)',
```

APPENDIX I. PATTERN 1 BEAMFORMING RESULTS WITH $R = 10$ CM

The following MATLAB code plots the result in Figure 31 for the four-element array on the cylindrical-shaped surface with $r = 10$ cm.

```
HFSS_case1 = [
-90 -7.699761969
-89 -7.564023248
-88 -7.434390267
-87 -7.311006445
-86 -7.194050404
-85 -7.083737911
-84 -6.980323733
-83 -6.884103455
-82 -6.795415307
-81 -6.714642048
-80 -6.642212958
-79 -6.578605991
-78 -6.524350141
-77 -6.480028094
-76 -6.446279242
-75 -6.423803147
-74 -6.413363567
-73 -6.41579317
-72 -6.4319991
-71 -6.462969585
-70 -6.509781831
-69 -6.573611491
-68 -6.655744089
-67 -6.757588855
-66 -6.88069557
-65 -7.026775183
-64 -7.197725176
-63 -7.39566097
-62 -7.622955065
-61 -7.882286212
-60 -8.176701667
-59 -8.509696778
-58 -8.885317754
-57 -9.308295847
-56 -9.784224668
-55 -10.31979745
-54 -10.92312866
-53 -11.60419522
-52 -12.37544774
-51 -13.25265984
-50 -14.25609237
-49 -15.41199742
-48 -16.75416708
-47 -18.32380708
-46 -20.16005841
-45 -22.24951431
-44 -24.32903914
-43 -25.49448879
-42 -24.82262163
-41 -23.00344403
-40 -21.06609096
-39 -19.36958709
-38 -17.95525401
-37 -16.79072565
-36 -15.83698042
-35 -15.06186288
-34 -14.44104081
-33 -13.95650586
-32 -13.59490328
-31 -13.3460824
-30 -13.20179985
-29 -13.15438343
-28 -13.19510085
-27 -13.31191438
-26 -13.48628237
-25 -13.6888766
-24 -13.87496951
-23 -13.98240368
-22 -13.9379636
-21 -13.67716455
-20 -13.17142386
-19 -12.44240142
-18 -11.5498446
-17 -10.5645656
-16 -9.547356975
-15 -8.541392021
-14 -7.573633042
-13 -6.659224041
-12 -5.805670381
-11 -5.015892174
-10 -4.2902119
-9 -3.627573663
```

```
-8 -3.026265914
-7 -2.484339792
-6 -1.999846042
-5 -1.570965733
-4 -1.196079808
-3 -0.873804252
-2 -0.603006768
-1 -0.382814463
0 -0.212618331
1 -0.092078148
2 -0.021130246
3 0
4 -0.029220692
5 -0.109660463
6 -0.242559375
7 -0.429579115
8 -0.672868605
9 -0.975149772
10 -1.339828961
11 -1.771140835
12 -2.274332741
13 -2.855897222
14 -3.523855391
15 -4.288075554
16 -5.16055625
17 -6.155446467
18 -7.288139835
19 -8.571568602
20 -10.00454809
21 -11.53928458
22 -13.00633881
23 -14.0221619
24 -14.1299739
25 -13.30767033
26 -11.99815525
27 -10.59935266
28 -9.292619484
29 -8.13047248
30 -7.115129036
31 -6.233277233
32 -5.468659268
33 -4.806111359
34 -4.232581904
35 -3.737144249
36 -3.310718665
37 -2.945749172
38 -2.635912593
39 -2.37587766
40 -2.161112056
41 -1.987729786
42 -1.852370759
43 -1.752105529
44 -1.684359542
45 -1.646852471
46 -1.637549228
47 -1.654620102
48 -1.696408
49 -1.761401322
50 -1.848211282
51 -1.955552791
52 -2.082228216
53 -2.227113464
54 -2.389145993
55 -2.567314397
56 -2.760649331
57 -2.968215546
58 -3.189104903
59 -3.422430203
60 -3.667319763
61 -3.922912623
62 -4.188354348
63 -4.462793341
64 -4.745377655
65 -5.035252241
66 -5.331556637
67 -5.633423047
68 -5.939974842
69 -6.250325439
70 -6.563577601
71 -6.878823148
72 -7.195143115
73 -7.511608391
74 -7.827280869
75 -8.141215161
76 -8.452460935
77 -8.760065915
78 -9.063079609
79 -9.360557809
80 -9.651567894
81 -9.935194949
82 -10.21054869
83 -10.47677114
84 -10.73304501
85 -10.97860253
86 -11.21273473
87 -11.43480076
88 -11.64423711
89 -11.84056632
90 -12.02340496
];
```

```
Meas_case1 = [
90 0.004545436
89 0.004464498
88 0.004526341
87 0.004910909
86 0.004067614
85 0.005817755
84 0.005825197
83 0.006445865
82 0.006501561
81 0.007080818
80 0.007262584
79 0.00761187
78 0.00789004
77 0.008343151
76 0.008721915
75 0.009062382
74 0.009427128
73 0.009622799
72 0.009968881
71 0.00995106
70 0.010034053
69 0.010162873
68 0.010294259
67 0.010312637
66 0.010649695
65 0.010570473
64 0.010934366
63 0.011133205
62 0.011294904
61 0.011622956
60 0.011960145
59 0.012478186
58 0.012750105
57 0.012994505
56 0.013315627
55 0.013626754
54 0.013973309
53 0.014317978
52 0.014453817
51 0.014323939
50 0.014455198
49 0.01465662
48 0.014038663
47 0.013910302
46 0.013495861
45 0.012968986
44 0.012795708
43 0.012587023
42 0.012087275
41 0.011862028
40 0.011228422
39 0.01059415
38 0.009910818
37 0.009523169
36 0.008775441
35 0.0081411
34 0.007368806
33 0.006526734
32 0.005590426
31 0.004585323
30 0.003675491
29 0.002962126
28 0.002460204
27 0.002772513
26 0.003194645
25 0.004469649
24 0.005630082
23 0.00686139
22 0.008172521
21 0.009604312
20 0.010871003
19 0.012287593
18 0.013409727
17 0.014632399
16 0.015709705
15 0.016744565
14 0.017705689
13 0.018622878
12 0.019176216
11 0.019773593
10 0.020469282
9 0.020555565
8 0.02100079
7 0.021161929
6 0.021129188
5 0.021180038
4 0.021060374
3 0.020827805
2 0.020466958
1 0.02022434
0 0.019801271
-1 0.019182594
-2 0.01854999
-3 0.018083733
-4 0.017308595
-5 0.016569023
-6 0.015420659
-7 0.014747736
-8 0.01363605
-9 0.012685019
-10 0.011430154
-11 0.010535332
-12 0.009340008
```

```
-13 0.008265292
-14 0.00685573
-15 0.005911287
-16 0.00463212
-17 0.003643911
-18 0.002808856
-19 0.002152698
-20 0.001802117
-21 0.001636443
-22 0.001682192
-23 0.001799374
-24 0.002295019
-25 0.002586932
-26 0.002380521
-27 0.002575293
-28 0.002479216
-29 0.00241215
-30 0.0023115
-31 0.002053073
-32 0.0019886
-33 0.001791012
-34 0.001680781
-35 0.001624608
-36 0.001629223
-37 0.00169876
-38 0.001854049
-39 0.002061852
-40 0.00223112
-41 0.002375037
-42 0.002585751
-43 0.002783723
-44 0.003093205
-45 0.003357823
-46 0.003542157
-47 0.004051943
-48 0.004240799
-49 0.004804199
-50 0.00529016
-51 0.005696588
-52 0.006169244
-53 0.006601265
-54 0.007174793
-55 0.007497103
-56 0.007881968
-57 0.008464659
-58 0.008916704
-59 0.009224472
-60 0.009712913
-61 0.009840812
-62 0.010075482
-63 0.010125484
-64 0.010154113
-65 0.010264028
-66 0.010151358
-67 0.010009696
-68 0.010140778
-69 0.009877523
-70 0.009765531
-71 0.009617377
-72 0.0093022
-73 0.009099893
-74 0.008804995
-75 0.008280619
-76 0.007792759
-77 0.007228552
-78 0.006702811
-79 0.006214599
-80 0.005866145
-81 0.005683734
-82 0.005323478
-83 0.005123008
-84 0.004743729
-85 0.004396232
-86 0.004417535
-87 0.004243534
-88 0.004394516
-89 0.004095377
-90 0.004152222
];

%close all;
clear all;
clc;
format short
N =4;
c = 3e8;
```

```
f = 2.4e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_s = 0*(pi/180);  % Angle from x-axis = 90 deg or angle
%from z-axis = 0 deg
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);        %receive angle
th1 = -30*(pi/180); %-20*(pi/180); % first interferer angle
th2 = 30*(pi/180); %10*(pi/180);   % second interferer angle
th3 = 40*(pi/180); %30*(pi/180);   % third interferer angle

% Elements locations on curved conformal surface
%#################################################################
r = 0.2032;  % For 8 inches radius sphere
theta1 = d/r;
theta2 = d/(2*r);
phi1 = pi/2 - (theta1 + theta2);
phi2 = phi1 + theta1;
phi3 = phi2;
phi4 = phi1;
phi_n = [phi1;phi2;phi3;phi4];
for m = 1:N/2
   x_n(m) = -r*cos(phi_n(m));
end
```

```
for m = (N/2+1):N
   x_n(:,m) = r*cos(phi_n(m));
   % x_n = [ -0.0896    -0.0311    0.0311    0.0896]
end
for m = 1:N
   z_n(m) = r*sin(phi_n(m));
   % z_n = [0.1578    0.1788    0.1788    0.1578]
end
%##################################################################
for m = 1: N
    % h_n = [ 0.0927    0.0312    0.0312    0.0927]
    h_n(m) = sqrt((0-x_n(m))^2 + (r-z_n(m))^2);
    phi_inv(m) = asin((r-z_n(m))/h_n(m));
end
theta_b = phi_inv;
theta_b_degrees = theta_b*180/pi
% For theta_s = 0, the wave front hits all elements with negative
% phase delays
% with respect to origin
psi1_SOI = exp(-1j*k*h_n(1)*sin(theta_b(1) + abs(th0)))*
              cos(pi/2-phi_n(1)+th0);
psi2_SOI = exp(-1j*k*h_n(2)*sin(theta_b(2) + abs(th0)))*
              cos(pi/2-phi_n(2)+th0);
psi3_SOI = exp(-1j*k*h_n(3)*sin(-theta_b(3) + abs(th0)))*
              cos(pi/2-phi_n(3)-th0);
psi4_SOI = exp(-1j*k*h_n(4)*sin(-theta_b(4) + abs(th0)))*
```

```
              cos(pi/2-phi_n(4)-th0);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].'; % a0
% For theta_s = theta_SNOI =  -30 < 0 and > theta_b, the
% wave front hits left elements 1,2
% first and right elements 3,4 later with respect to origin
psi1_SNOI1 = exp(1j*k*h_n(1)*sin(-theta_b(1) + abs(th1)))*
                cos(pi/2-phi_n(1)+th1);
psi2_SNOI1 = exp(1j*k*h_n(2)*sin(-theta_b(2) + abs(th1)))*
                cos(pi/2-phi_n(2)+th1);
psi3_SNOI1 = exp(-1j*k*h_n(3)*sin(theta_b(3) + abs(th1)))*
                cos(pi/2-phi_n(3)-th1);
psi4_SNOI1 = exp(-1j*k*h_n(4)*sin(theta_b(4) + abs(th1)))*
                cos(pi/2-phi_n(4)-th1);
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI = 30 > theta_b, the wave front hits right
% elements 3,4 first and left elements 1,2 later with respect
% to origin
psi1_SNOI2 = exp(-1j*k*h_n(1)*sin(theta_b(1) + abs(th2)))*
                cos(pi/2-phi_n(1)+th2);
psi2_SNOI2 = exp(-1j*k*h_n(2)*sin(theta_b(2) + abs(th2)))*
                cos(pi/2-phi_n(2)+th2);
psi3_SNOI2 = exp(1j*k*h_n(3)*sin(-theta_b(3) + abs(th2)))*
                cos(pi/2-phi_n(3)-th2);
psi4_SNOI2 = exp(1j*k*h_n(4)*sin(-theta_b(4) + abs(th2)))*
                cos(pi/2-phi_n(4)-th2);
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
```

```
% For SNOI = 40  > theta_b
psi1_SNOI3 = exp(-1j*k*h_n(1)*sin(theta_b(1) + abs(th3)))*
                cos(pi/2-phi_n(1)+th3);
psi2_SNOI3 = exp(-1j*k*h_n(2)*sin(theta_b(2) + abs(th3)))*
                cos(pi/2-phi_n(2)+th3);
psi3_SNOI3 = exp(1j*k*h_n(3)*sin(-theta_b(3) + abs(th3)))*
                cos(pi/2-phi_n(3)-th3);
psi4_SNOI3 = exp(1j*k*h_n(4)*sin(-theta_b(4) + abs(th3)))*
                cos(pi/2-phi_n(4)-th3);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3

A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
w = u/A;
%######################################
% Z parameters from HFSS at f = 2.4729 GHz
% Self Terms
 Z11 = -37.87-16.15i;
% Mutual Terms
Z12 = 0.45+3.94i;
Z13 = -1.355+1.923i;
Z14 = 0.8+0.186i;
Z21 = Z12;
Z23 = -2.63+3.77i;
Z24 = -1.25+1.82i;
Z31 = Z13;
```

```
Z32 = Z23;
Z34 = 0.844+3.856i;
Z41 = Z14;
Z42 = Z24;
Z43 = Z34;
Z22 = Z11;
Z33 = Z11;
Z44 = Z11;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0;
    Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I ;  %Normalized coupling impedance matrix
Ac = Zc\A;   %inv(Zc)*A;
wc = u/Ac;
 w_mag_norm = abs(w)./max(abs(w))
% w_mag_dB = 20*log10(w_mag_norm);
 w_phase_deg = angle(w).*(180/pi)
 wc_mag_norm = abs(wc)./max(abs(wc))
% wc_mag_dB = 20*log10(w_mag_norm);
 wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
e_l1 = cos(pi/2 - phi_n(1)+theta);
e_l2 = cos(pi/2 - phi_n(2)+theta);
```

```
e_r3 = cos(pi/2 - phi_n(3)-theta);
e_r4 = cos(pi/2 - phi_n(4)-theta);

 u1 = sin(theta);
 u2 = cos(theta);
 figure
 % No mutual ccoupling
  AP = w(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           w(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           w(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           w(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));
 % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(j.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           wc(2).*e_l2.*exp(j.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           wc(3).*e_r3.*exp(j.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           wc(4).*e_r4.*exp(j.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
APc_norm = abs(APc)./max(abs(APc));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
```

```
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
Meas_case1_norm = abs(Meas_case1(:,2))./max((Meas_case1(:,2)));
plot(-Meas_case1(:,1),20*log10(Meas_case1_norm),'k.');
hold on
plot(-HFSS_case1(:,1),HFSS_case1(:,2),'kx');
xlabel('\theta(deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)','Analytical (with coupling)'
       ,'Measured','HFSS')
%plot(theta*180/pi,e_l2)
%plot(x_n,z_n,'o')
```

APPENDIX J. PATTERN 2 BEAMFORMING RESULTS WITH $R = 10$ CM

The following MATLAB code plots the result in Figure 32 for the four-element array on the cylindrical-shaped surface with $r = 10$ cm.

```
HFSS_case2 = [
-90 -17.49781058
-89 -17.33444117
-88 -17.14637871
-87 -16.93335302
-86 -16.69541578
-85 -16.43294503
-84 -16.14663485
-83 -15.83747151
-82 -15.50669831
-81 -15.15577243
-80 -14.78631715
-79 -14.40007325
-78 -13.99885239
-77 -13.58449489
-76 -13.15883351
-75 -12.72366394
-74 -12.28072208
-73 -11.83166779
-72 -11.37807435
-71 -10.9214228
-70 -10.4631001
-69 -10.00440047
-68 -9.546528749
-67 -9.090605473
-66 -8.637672823
-65 -8.188701158
-64 -7.744595766
-63 -7.306203592
-62 -6.874319793
-61 -6.449694006
-60 -6.03303628
-59 -5.625022643
-58 -5.2263003
-57 -4.837492467
-56 -4.459202888
-55 -4.092020031
-54 -3.73652103
-53 -3.393275376
-52 -3.062848419
-51 -2.745804699
-50 -2.442711145
-49 -2.154140182
-48 -1.880672784
-47 -1.622901497
-46 -1.381433494
-45 -1.156893696
-44 -0.949928009
-43 -0.761206738
-42 -0.591428246
-41 -0.441322934
-40 -0.311657632
-39 -0.203240513
-38 -0.116926654
-37 -0.053624409
-36 -0.014302777
-35 0
-34 -0.011833677
-33 -0.051012758
-32 -0.118851859
-31 -0.216788477
-30 -0.346403834
-29 -0.509448308
-28 -0.707872676
-27 -0.943866838
-26 -1.21990821
-25 -1.538822774
-24 -1.903862883
-23 -2.318807492
-22 -2.78809281
-21 -3.31698473
-20 -3.911809387
-19 -4.580265583
-18 -5.331853568
-17 -6.178469862
-16 -7.135236483
-15 -8.221646257
-14 -9.463070918
-13 -10.89241994
-12 -12.55054264
-11 -14.47888224
-10 -16.67697975
-9 -18.92745642
```

```
-8 -20.39121359
-7 -20.02468666
-6 -18.36981915
-5 -16.52829906
-4 -14.91775856
-3 -13.59299958
-2 -12.52188705
-1 -11.6631694
0 -10.98236111
1 -10.45311972
2 -10.05581138
3 -9.775915025
4 -9.602765962
5 -9.528650585
6 -9.548175123
7 -9.657833859
8 -9.855720415
9 -10.14134134
10 -10.51550074
11 -10.98022666
12 -11.53870172
13 -12.19513372
14 -12.95443979
15 -13.8214798
16 -14.7992807
17 -15.88509338
18 -17.06204947
19 -18.28300509
20 -19.44533052
21 -20.3726575
22 -20.85792861
23 -20.8006559
24 -20.29995876
25 -19.56295133
26 -18.76696444
27 -18.01562651
28 -17.35686967
29 -16.8082058
30 -16.37297967
31 -16.04877505
32 -15.83138932
33 -15.71661756
34 -15.70100441
35 -15.78211198
36 -15.95854443
37 -16.22980928
38 -16.59599157
39 -17.05711279
40 -17.61189413
41 -18.25539024
42 -18.97457386
43 -19.74058882
44 -20.49698875
45 -21.14775103
46 -21.56083753
47 -21.61200935
48 -21.25937509
49 -20.57410708
50 -19.68761529
51 -18.72076021
52 -17.75401281
53 -16.83113673
54 -15.97202565
55 -15.18327613
56 -14.46473563
57 -13.81311291
58 -13.2238557
59 -12.69208804
60 -12.21305619
61 -11.78232279
62 -11.39583475
63 -11.04992971
64 -10.74131432
65 -10.46703141
66 -10.22442453
67 -10.01110374
68 -9.824914538
69 -9.663910366
70 -9.526328711
71 -9.41057053
72 -9.31518259
73 -9.238842338
74 -9.180344903
75 -9.138591897
76 -9.112581693
77 -9.10140094
78 -9.104217068
79 -9.120271608
80 -9.148874155
81 -9.189396843
82 -9.241269201
83 -9.303973309
84 -9.377039132
85 -9.460039976
86 -9.552587971
87 -9.65432951
88 -9.764940572
89 -9.884121837
90 -10.01159352
];
```

```
Meas_case2 = [
90 0.005379713
89 0.005327968
88 0.005229077
87 0.003620271
86 0.004187664
85 0.004173925
84 0.004268526
83 0.004502444
82 0.004915263
81 0.005444111
80 0.005821133
79 0.006578969
78 0.007222913
77 0.007649016
76 0.008149314
75 0.008780653
74 0.009357899
73 0.00961975
72 0.010212752
71 0.010096747
70 0.010118154
69 0.010137471
68 0.010149732
67 0.009660603
66 0.009282193
65 0.00877458
64 0.008377722
63 0.00780107
62 0.007018703
61 0.006564338
60 0.005564076
59 0.004653742
58 0.00388595
57 0.00281597
56 0.002056156
55 0.000888529
54 0.00033664
53 0.001061667
52 0.001633635
51 0.002647827
50 0.003131348
49 0.003814634
48 0.004050629
47 0.004504728
46 0.004839674
45 0.005103696
44 0.005279312
43 0.005468995
42 0.005661991
41 0.005665414
40 0.005652798
39 0.005648166
38 0.005635435
37 0.005525267
36 0.005384371
35 0.005489622
34 0.005140715
33 0.004820596
32 0.004814068
31 0.004636606
30 0.004247092
29 0.004140805
28 0.00367055
27 0.00329899
26 0.003169944
25 0.003191874
24 0.003240534
23 0.003303634
22 0.00334314
21 0.003550584
20 0.004130215
19 0.004491379
18 0.004941333
17 0.005762901
16 0.006463127
15 0.007095413
14 0.007747393
13 0.008638175
12 0.009117954
11 0.009731563
10 0.010115502
9 0.01069468
8 0.011037761
7 0.01130935
6 0.011366059
5 0.011288745
4 0.011265297
3 0.010898233
2 0.01085648
1 0.010706383
0 0.010458171
-1 0.010220418
-2 0.009974416
-3 0.009091267
-4 0.008728578
-5 0.008304698
-6 0.007848829
-7 0.007431185
-8 0.007120622
-9 0.006503809
-10 0.006239231
-11 0.006087989
```

```
-12 0.005943051   -32 0.020789691   -52 0.017771216   -72 0.010723701
-13 0.006098258   -33 0.021085197   -53 0.017272996   -73 0.010371605
-14 0.006505539   -34 0.021159414   -54 0.016960113   -74 0.009855391
-15 0.007286123   -35 0.021264382   -55 0.0165132     -75 0.009127716
-16 0.007926386   -36 0.021447327   -56 0.015905832   -76 0.008233972
-17 0.009091223   -37 0.021565867   -57 0.015645732   -77 0.007344748
-18 0.010082952   -38 0.021559893   -58 0.015564153   -78 0.006732662
-19 0.011415235   -39 0.021461342   -59 0.015158958   -79 0.006115747
-20 0.012397931   -40 0.021452018   -60 0.014851089   -80 0.005596771
-21 0.013622857   -41 0.021348282   -61 0.014535192   -81 0.005156186
-22 0.014566106   -42 0.021152173   -62 0.014276737   -82 0.00488069
-23 0.015178976   -43 0.020945004   -63 0.0141666     -83 0.004560371
-24 0.016071535   -44 0.020704854   -64 0.013969016   -84 0.004626517
-25 0.016912983   -45 0.020405004   -65 0.013565928   -85 0.004740726
-26 0.017455902   -46 0.020162176   -66 0.013105516   -86 0.004847864
-27 0.0183507     -47 0.01994065    -67 0.012967344   -87 0.005299266
-28 0.018851073   -48 0.019234673   -68 0.012549181   -88 0.005503867
-29 0.019351066   -49 0.018858956   -69 0.01206181    -89 0.005443955
-30 0.019819691   -50 0.018638814   -70 0.011726477   -90 0.005751116
-31 0.02032424    -51 0.018242237   -71 0.011447511   ];

%close all;
clear all;
clc;
format short
N =4;
c = 3e8;
```

```
f = 2.4e9;
lambda = c/f;
d = lambda/2;
k = (2*pi)/lambda;
theta_s = 40*(pi/180); % Angle from x-axis = 90 deg or angle
% from z-axis
                       % = 0 deg
% Four elements antenna array
% Projections on wavefront
th0 = theta_s; %0*(pi/180);         %receive angle
th1 = -45*(pi/180); %-20*(pi/180);  % first interferer angle
th2 = -25*(pi/180); %10*(pi/180);   % second interferer angle
th3 = 10*(pi/180); %30*(pi/180);    % third interferer angle

% Elements locations on curved conformal surface
%#################################################################
r = 0.2032; % For 8 inches = 0.2032 m radius sphere
theta1 = d/r;
theta2 = d/(2*r);
phi1 = pi/2 - (theta1 + theta2);
phi2 = phi1 + theta1;
phi3 = phi2;
phi4 = phi1;
phi_n = [phi1;phi2;phi3;phi4];
for m = 1:N/2
   x_n(m) = -r*cos(phi_n(m));
```

```
end
for m = (N/2+1):N
   x_n(:,m) = r*cos(phi_n(m));
   % x_n = [ -0.0896    -0.0311    0.0311    0.0896]
end
for m = 1:N
   z_n(m) = r*sin(phi_n(m));
   % z_n = [0.1578    0.1788    0.1788    0.1578]
end
%##################################################################
for m = 1: N
% h_n = [ 0.0927;0.0312;0.0312;0.0927]
    h_n(m) = sqrt((0-x_n(m))^2 + (r-z_n(m))^2);
    phi_inv(m) = asin((r-z_n(m))/h_n(m));
end
theta_b = phi_inv;
theta_b_degrees = theta_b*180/pi
% theta_b = [13.2172  4.4057  4.4057  13.2172]

% For theta_s = 40 > 0, theta_b, the wave front hits right elements
% 3,4 first and left elements 1,2 later with respect to origin
psi1_SOI = exp(-1j*k*h_n(1)*sin(theta_b(1) + abs(th0)))*
              cos(pi/2-phi_n(1)+th0);
psi2_SOI = exp(-1j*k*h_n(2)*sin(theta_b(2) + abs(th0)))*
              cos(pi/2-phi_n(2)+th0);
psi3_SOI = exp(1j*k*h_n(3)*sin(-theta_b(3) + abs(th0)))*
```

```
              cos(pi/2-phi_n(3)-th0);
psi4_SOI = exp(1j*k*h_n(4)*sin(-theta_b(4) + abs(th0)))*
              cos(pi/2-phi_n(4)-th0);
SOI = [psi1_SOI psi2_SOI psi3_SOI psi4_SOI].'; % a0
% For theta_s = theta_SNOI =  -45 < 0 and > theta_b,
% the wave front hits left elements 1,2 first and right
% elements 3,4 later with respect to  the origin
psi1_SNOI1 = exp(1j*k*h_n(1)*sin(-theta_b(1) + abs(th1)))*
                cos(pi/2-phi_n(1)+th1);
psi2_SNOI1 = exp(1j*k*h_n(2)*sin(-theta_b(2) + abs(th1)))*
                cos(pi/2-phi_n(2)+th1);
psi3_SNOI1 = exp(-1j*k*h_n(3)*sin(theta_b(3) + abs(th1)))*
                cos(pi/2-phi_n(3)-th1);
psi4_SNOI1 = exp(-1j*k*h_n(4)*sin(theta_b(4) + abs(th1)))*
                cos(pi/2-phi_n(4)-th1);
SNOI1 = [psi1_SNOI1 psi2_SNOI1 psi3_SNOI1 psi4_SNOI1].'; % a1
% For SOI, SNOI < 0
psi1_SNOI2 = exp(1j*k*h_n(1)*sin(-theta_b(1) + abs(th2)))*
                cos(pi/2-phi_n(1)+th2);
psi2_SNOI2 = exp(1j*k*h_n(2)*sin(-theta_b(2) + abs(th2)))*
                cos(pi/2-phi_n(2)+th2);
psi3_SNOI2 = exp(-1j*k*h_n(3)*sin(theta_b(3) + abs(th2)))*
                cos(pi/2-phi_n(3)-th2);
psi4_SNOI2 = exp(-1j*k*h_n(4)*sin(theta_b(4) + abs(th2)))*
                cos(pi/2-phi_n(4)-th2);
SNOI2 = [psi1_SNOI2 psi2_SNOI2 psi3_SNOI2 psi4_SNOI2].'; % a2
```

```
% For SNOI = 10  > 0
psi1_SNOI3 = exp(-1j*k*h_n(1)*sin(theta_b(1) + abs(th3)))*
                cos(pi/2-phi_n(1)+th3);
psi2_SNOI3 = exp(-1j*k*h_n(2)*sin(theta_b(2) + abs(th3)))*
               cos(pi/2-phi_n(2)+th3);
psi3_SNOI3 = exp(1j*k*h_n(3)*sin(-theta_b(3) + abs(th3)))*
               cos(pi/2-phi_n(3)-th3);
psi4_SNOI3 = exp(-1j*k*h_n(4)*sin(theta_b(4) - abs(th3)))*
               cos(pi/2-phi_n(4)-th3);
SNOI3 = [psi1_SNOI3 psi2_SNOI3 psi3_SNOI3 psi4_SNOI3].'; % a3

A = [SOI SNOI1 SNOI2 SNOI3];
u = [1 0 0 0];
w = u/A;  % Weights with out including effect of mutual coupling

%######################################
% Z parameters from HFSS at f = 2.4729 GHz
% Self Terms
Z11 = -30.95-25.57i ;
% Z22 = -37.87-16.15i;
% Z33 = -41.18-5.56i;
% Z44 = -5.11-39.33i;
% Mutual Terms
Z12 = 0.45+3.94i;
Z13 = -1.355+1.923i;
Z14 = 0.8+0.186i;
```

```
Z21 = Z12;
Z23 = -2.63+3.77i;
Z24 = -1.25+1.82i;
Z31 = Z13;
Z32 = Z23;
Z34 = 0.844+3.856i;
Z41 = Z14;
Z42 = Z24;
Z43 = Z34;
 Z22 = Z11;
 Z33 = Z11;
 Z44 = Z11;
Z0 = 50;
% Normalized Impdance Matrix
Z = [Z11/Z0 Z12/Z0 Z13/Z0 Z14/Z0;Z21/Z0 Z22/Z0 Z23/Z0 Z24/Z0;
    Z31/Z0 Z32/Z0 Z33/Z0 Z34/Z0;Z41/Z0 Z42/Z0 Z43/Z0 Z44/Z0];
I = eye(4);
Zc = Z + I ;  %Normalized coupling impedance matrix
Ac = Zc\A;   %inv(Zc)*A;
wc = u/Ac;
%w = Z\w.'; %inv(Z)*w.';
w_mag_norm = abs(w)./max(abs(w))
%w_mag_dB = 20*log10(w_mag_norm);
w_phase_deg = angle(w).*(180/pi)
wc_mag_norm = abs(wc)./max(abs(wc))
%wc_mag_dB = 20*log10(wc_mag_norm);
```

```
wc_phase_deg = angle(wc).*(180/pi)
% Array factor
theta = (pi/180)*(-90:180/628:90);
%theta = (pi/180)*(-90:2:90);
e_l1 = cos(pi/2 - phi_n(1)+theta);
e_l2 = cos(pi/2 - phi_n(2)+theta);
e_r3 = cos(pi/2 - phi_n(3)-theta);
e_r4 = cos(pi/2 - phi_n(4)-theta);

 u1 = sin(theta);
 u2 = cos(theta);

 %w = [1,1,1,1];
 % No mutual ccoupling
 figure
  AP = w(1).*e_l1.*exp(1i.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           w(2).*e_l2.*exp(1i.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           w(3).*e_r3.*exp(1i.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           w(4).*e_r4.*exp(1i.*k.*(x_n(4).*u1 + z_n(4).*u2));
 % Effects of mutual coupling
     APc = wc(1).*e_l1.*exp(1i.*k.*(x_n(1).*u1 + z_n(1).*u2)) + ...
           wc(2).*e_l2.*exp(1i.*k.*(x_n(2).*u1 + z_n(2).*u2))+ ...
           wc(3).*e_r3.*exp(1i.*k.*(x_n(3).*u1 + z_n(3).*u2))+ ...
           wc(4).*e_r4.*exp(1i.*k.*(x_n(4).*u1 + z_n(4).*u2));

AP_norm = abs(AP)./max(abs(AP));
```

```
APc_norm = abs(APc)./max(abs(APc));
set(0,'DefaultAxesColorOrder',[0 0 0], ...
    'DefaultLineMarkerSize', 8, ...
    'DefaultAxesFontSize', 16, ...
    'DefaultTextFontSize', 16, ...
    'DefaultLineLineWidth',2);
plot(theta*180/pi,20*log10(AP_norm),'k-');
hold on
plot(theta*180/pi,20*log10(APc_norm),'k--');
hold on
Meas_case2_norm = abs(Meas_case2(:,2))./max((Meas_case2(:,2)));
plot(-Meas_case2(:,1),20*log10(Meas_case2_norm),'k.');
hold on
plot(-HFSS_case2(:,1),HFSS_case2(:,2),'kx');

xlabel('\theta(deg)')
ylabel('|E_{\phi}| (dB)')
axis([-90 90 -50 0])
legend('Analytical (without coupling)','Analytical (with coupling)'
       ,'Measured','HFSS')
%plot(theta*180/pi,e_l2)
%plot(x_n,z_n,'o')
```