Antenna array beam pattern design for 5G millimeter wave applications

Ioannis Tsiknas

SID: 3307170005

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Communications and Cybersecurity

DECEMBER 2018

THESSALONIKI – GREECE

Antenna array beam pattern design for 5G millimeter wave applications

Ioannis Tsiknas

SID: 3307170005

Supervisor: Prof. Marios Gatzianas

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) Communications and Cybersecurity

DECEMBER 2018

THESSALONIKI – GREECE

Abstract

This thesis was written as a part of the MSc in Communications and Cybersecurity at the International Hellenic University. Its aim is to examine antenna array radiation pattern synthesis in the millimeter wave bands, for use in future 5G applications. After a literature review in 5G, millimeter waves, antenna arrays and beamforming, two pattern synthesis algorithms were developed in Matlab, in order to synthesize radiation patterns and review the results. One algorithm was based on the solution of a linear system, while the other was based on the Fourier transform method. After the comparison of the results, it is concluded that the Fourier transform algorithm is performing better and as expected than the linear system algorithm.

I would like to express my sincere gratitude to my supervisor, Dr. Marios Gatzianas, for his guidance and advice, as this work would not be possible without him.

Ioannis Tsiknas

5 December 2018

Contents

1 Introduction

Initially, cellular wireless networks were able to support only voice and text messages. Gradually, they started to support data as well. 3G was a major leap in wireless data connectivity, allowing mobile devices to smoothly browse the web and stream videos and music. When 4G was commercially deployed, data rates were increased tremendously, supporting TV live stream services. The upcoming 5G intends to evolve the existing technologies, in order to achieve better coverage, larger capacity, low latency, while maintaining energy efficiency. High capacity cells with antenna array communication systems is one of the keys in meeting these goals, and so, antenna array analysis and designs will be the main focus of this thesis.

This work aims to design antenna arrays and construct their radiation patterns in the 60 GHz millimeter wave band, which is expected to be employed in the forthcoming 5G networks, by using as input a desired radiation pattern. This procedure is performed in Matlab, with the assistance of the Antenna Toolbox, by using two different pattern synthesis algorithms. The first algorithm is based on the solution of a linear system and the second one on the Fourier transform method. Comparative analysis on both x-y and x-z planes, showed that the algorithm based on the Fourier transform method demonstrated better and more accurate results than the algorithm based on the solution of a linear system.

The outline of the thesis is as follows. Chapter 2 contains a literature review on aspects of 5G, millimeter waves, antenna arrays and beamforming. Chapter 3 contains descriptions of the functions used in Matlab, as well as an illustration of the two pattern synthesis algorithms that are used. Chapter 4 contains a presentation of the numerical results of the Matlab simulations, a comparison between the two algorithms and further examination of the Fourier transform algorithm. Lastly, Chapter 5 contains the final conclusions of the thesis and possible future work.

2 Literature Review

2.1 A very brief overview of 5G

5G is going to embrace existing technologies and add new ones on top of them. As Long Term Evolution (LTE) cellular networks cannot underpin the increased traffic capacity, 5G has to overcome a number of challenges, in order to improve the current state of communications. Data rates of 1-10 Gbps, 1ms round trip latency, massive device connectivity, high availability and coverage for every device, reasonable Quality of Service (QoS) in densely populated areas, energy efficiency for both the transmitter and the receiver and incorporation of Internet of Things (IoT) devices into mobile networks (which has its own challenges to cope with), are the core requirements of the 5G ecosystem. Millimeter waves and beamforming, two significant factors for enabling 5G, will not be discussed in this overview, as they will be examined separately in the following sections [1].

2.1.1 Device-to-Device & Machine-to-Machine

The radio access network will be more decentralized, compared to previous architectures, in order to allow Device-to-Device (D2D) and Machine-to-Machine (M2M) communications. The nature of D2D allows mobile devices to connect/communicate to each other directly without the need of a cellular base station. As a result, coverage is increased and the number of wireless hops is reduced; therefore, latency, transmission power and control signaling are decreased, allowing for energy/battery saving and there is higher potential for scalability. Data caching for frequently requested data at mobile devices or edge nodes will greatly reduce the delay. In addition, M2M communications are expected to serve a vast number of devices in real time with low delay and high reliability [1, 2]. For example, different sensors, such as temperature, humidity, wind speed etc., can gather data and send them to a controller, which will determine the weather and inform any interested party in real time.

2.1.2 Massive MIMO (Multiple Input Multiple Output)

Massive MIMO [3] is a technology that scales up MIMO [4], utilizing a large number of antennas to multiplex outgoing data for several users on the same time/frequency resource, providing energy efficiency and increased data rates, as more antennas are being used compared to conventional MIMO. This also boosts the robustness of the network, rendering it sturdier against fading or failure of one or more antennas. The resilience and fading tolerance are a result of the combined operation of antennas. However, one limitation, namely Pilot Contamination, arises. As the maximum number of orthogonal pilot sequences that can be employed per cell is upper-bounded by the duration of the coherence interval divided by the channel delay spread, these pilot sequences must eventually be reused in different cells, leading to interference during the pilot-assisted channel estimation procedure. There are techniques to mitigate this problem, some of which are still under development. Another limitation is that wireless channel responses, between the base station and each mobile device, are diverse. Massive MIMO heavily relies on this fact, which is known as the Favorable Propagation assumption. Although, having a large number of base station antennas than users can stabilize the performance. Despite the two aforementioned limitations, which are considered manageable, there are other challenges that require further research, such as fast and distributed signal processing algorithms and the accompanying hardware (include the complete RF chains for Digital to Analog Converters (DACs), oscillators and power amplifiers) for multiple antennas. Low cost hardware may come with deficiencies, for example unacceptable noise isolation. Additional parameters and procedures must be considered in order to construct a model which describes accurately the fundamental nature and behavior of the channel; for example, Time Division Duplex (TDD) requires the occasional reciprocity calibration. Furthermore, each network may be deployed differently depending on current needs and the characteristics of each channel [1, 5].

2.1.3 Centralized Architecture

A centralized baseband architecture allows the separation of base stations (performing Digital Signal Processing and resource allocation) from the Remote Radio Heads (RRHs) through virtualization and centralization. Resources are gathered in a pool of Baseband Units (BBU pool), which is connected with the base stations. This architecture, also known as Cloud Radio Access Network (C-RAN), improves coverage, mobility, equipment and operation costs and offers energy efficiency. The RRHs are connected

to the BBU pool through optical fibers. RRHs are equipped with transceiver components, amplifiers and duplexers; thus, they are able to handle digital processing, analog to digital conversions, power amplifications and filtering [1].

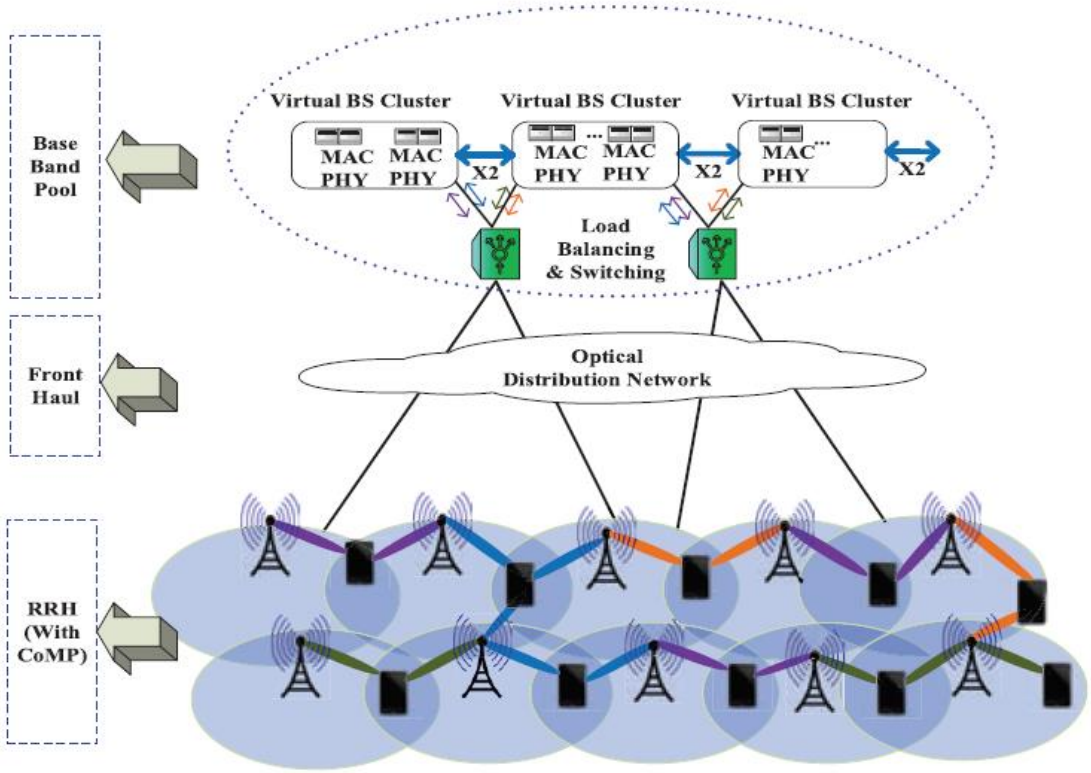

Figure 1: C-RAN architecture [1].

2.1.4 Software-Defined Networking

Software-Defined Networking (SDN) separates the control plane from the data plane. The functions of the control plane are implemented in a software-based controller, which can be used to guide and adjust the behavior of the data plane nodes. SDN has successfully increased performance in wired networks and has been implemented, up to a point, in Radio Access Networks (RANs) [6]. Thus, potentially upgrading to Cloud RANs (C-RANs), by adding an SDN controller to the fronthaul network (i.e. the part of the network between the BBU pool and the RRHs), will offer the increased flexibility and network slicing required by the envisioned 5G applications. C-RANs, by centralizing the signal processing in a single location, also facilitate Coordinated Multipoint (CoMP), which enables base stations to send data to a mobile device by cooperating with each other, therefore mitigating interference.

Self-Organizing Networking (SON) can be seen as an application of SDN for wireless networks. SON includes functions such as self-configuration, for example, management

of radio network parameters and software/firmware updates, self-optimization, for example, load balancing, which allows the sharing of traffic for better performance when needed, and self-healing, for example if a base station fails, then the neighboring base stations will adjust in order to increase their coverage.

Software-Defined Fronthaul (SDF) offers a lot of capabilities, such as effortless handovers, power saving by not using every Baseband Unit in the BBU pool and allows multiple operators to use the same fronthaul network, as SDF does not implement any operator specific standard. Similarly to SDN, SDF also requires a resource manager, as a controller, to compute the fronthaul configurations for the current traffic and user profiles, and a switching element to disseminate through the configuration data in real time.

Of course, there are challenges to overcome in order to implement SDF commercially, such as low latency requirements, non-standard communication protocols and heterogeneity of interfaces, i.e. the mixing of wireless, optical fiber and copper links, and the type of switching (electrical or optical) [1, 6].

2.1.5 Heterogeneous Networks

Heterogenous Networks (HetNets) were first proposed in 4G, but the architecture was not able to support them. HetNets will integrate different types of networks, such as Wi-Fi networks and cellular Radio Access Networks (RANs), while using different carrier frequencies and Radio Access Technologies (RATs). This will result in an outdoor to indoor and vice versa communication handled by individual nodes under the umbrella of 5G. HetNets can achieve the required high data rates and low latency, as more and more small cells can be deployed, causing a reduction in power consumption and an increase in spectrum efficiency [1, 7].

2.2 The role of millimeter waves in achieving high throughput in 5G

Current data rates are becoming obsolete due to the vast increase of mobile devices. 5G will take advantage of millimeter wave frequencies, in order to increase the available bandwidth (a bandwidth of 7 GHz is available in the 57-64 GHz ISM band) and at the same time decrease latency. Millimeter waves are considered a key element for the future of wireless communications, as they unlock a new range of frequencies 30-300 GHz, which are not currently used for cellular communications [1]. Apart from the in-

crease in capacity, high frequencies allow the use of smaller antenna dimensions. 5G will support high data rates and increased capacity by taking advantage of dense network cells. Network densification also brings some challenges, such as costly connectivity from base stations to the fronthaul/backhaul network using fiber, increase in inter-cell interference and more frequent handovers. The authors in [8] consider a point-to-multipoint (P2MP) architecture, which can deal with hardware costs and base stations communication latency issues. Using the same frequency band for multiplexing between the base station and the mobile device, as well as the base station and the backhaul link, allows us to reuse spectrum and hardware for the two links. In the first implementation of 5G, it is highly likely that operators will use already existing base stations, which will result in limited radio coverage. To address this issue there is a need of supplementary 5G base stations, which will also need to be connected to the core network, either by using fiber cables or wirelessly.

As with other bands of the frequency spectrum, millimeter waves also come with drawbacks and obstacles to overcome. The short wavelength limits the transmission range and penetration of the signal. Measurements at 28 and 38 GHz suggest that outdoor to indoor, or vice versa, communication must be handled by different nodes because of millimeter waves low penetration through common building materials. Signals at lower frequencies are able to penetrate more easily through buildings thus, penetration losses are higher for millimeter waves. In addition, smaller cell size, meaning basically more cells per area of interest, increases the aggregate capacity of the wireless network, while decreasing propagation losses, as the distance between a base station and a mobile device will not be more than 200 meters. Exceeding the distance of 200 meters highly impacts the power of the transmitted signal [9].

The millimeter wave propagation loss is given by the free space loss formula,

$$L = 32.4 + 20log_{10}f + 20log_{10}R \tag{2-1}$$

where L is the transmission loss in dB, f is the carrier frequency in GHz and R is the distance between the transmitter and the receiver in meters. From (2-1) we can deduce that path loss is increased for high carrier frequencies.

In [10], it is suggested that base stations deployment in millimeter waves must be more compact, approximately 100 to 500 meters, unlike previous architectures. Beams in the transmitter and the receiver have high directivity and high beamforming gain, so it is

easier to serve mobile devices at the cell's edge and also subdue co-channel interference. The geographic coverage area of the base stations creates a grid, compared with LTE coverage where base stations exhibit a hexagonal structure. As a result, mobile devices are able to connect to a large number of base stations, which allows high quality communication in any location within the covered area. Authors propose two approaches to implement millimeter waves along with the already deployed 4G systems. The first one is to establish both systems individually with the ability to switch between them. The second one (and the more likely to be used) is a hybrid operation in which 4G will handle control information exchange, while the millimeter wave system will be in charge of data communication.

Research and measurements have been performed to determine the effect of human blockage on millimeter wave communications. In [11], propagation measurements have been conducted in the presence of human activity for a 60 GHz channel. The environment where the measurements took place was a large indoor working place. The results showed that human activity plays a major role in propagation characteristics for millimeter waves communications, especially if the direct path is obscured. The authors in [12] present urban cellular and peer-to-peer RF wideband channel measurements using steerable antennas at 38 and 60 GHz. For the peer-to-peer channel, the experiments were conducted in an urban location, i.e. a pedestrian walkway. Measurements showed that Non Line of Sight (NLOS) links had higher path loss and RMS delay in comparison to line of sight (LOS) links. The 38 GHz channel resulted in higher RMS delay spreads than the 60 GHz channel, due to the fact that the environment had a lower free space path loss and there were more obstacles, which acted as scatterers. For the cellular channel, the experiments were conducted in a campus, also considered an urban location. 38 GHz LOS and partially obstructed LOS links had minimal RMS delay spread, while NLOS links featured higher RMS delay spread. NLOS problems can be solved by a system with the ability to determine the best combination of transmitter and receiver pointing angles. In this way, SNR and throughput will be increased and RMS delay spread will decline.

Another, experiment [13] shows that millimeter wave bands of 28, 38, 71-76 and 81-86 GHz have similar path loss behavior, which can be adjusted by the size of antenna arrays. The more elements an antenna array has, the lower will be the path loss, thus making the aforementioned bands comparable in terms of transmission losses.

For channel modelling, a number of parameters must be known, in order to have a clear picture of the channel, i.e. the number of scattering clusters, number of rays of each cluster, time of arrival of each ray, azimuth and elevation angles of departure and arrival. Millimeter wave channels also have additional propagation characteristics, which should be taken into consideration. These characteristics are free space path loss, reflection loss and diffraction loss. Outdoor and indoor measurements at 28 GHz frequency band are taken in [14]. The channel model seems to befall with the actual measurements, but further rectifications must be made to improve the model's accuracy.

2.3 Antenna Arrays

The small wave length of millimeter wave signals permits the creation of antenna arrays by placing many antenna elements next to each other in a specific way, occupying only a small amount of space. Antenna arrays have properties which can be manipulated to produce a desirable radiation pattern, helping with the proper pairing between the base station and the mobile device. Such properties are the number of individual elements, their type, e.g. dipoles, microstrip, horn antennas, the configuration between them, e.g. linear and rectangular, the spacing between them and their excitation amplitude and phase. Adaptive antenna arrays with several radiating elements can be designed. Beam steering can be enabled by giving a different weight to each radiating element, thus adjusting the amplitude and phase. As a result, research is focusing on directional narrow beam communications.

Authors in [15] have designed and analyzed a broadband multi-beam end-fire antenna array. The antenna has low cost and is easy to integrate, while its performance shows good signs in millimeter wave frequencies making it a candidate for 5G communications.

In [16], planar, circular and three-segment array configurations are examined and analyzed. Each array is formed using vertical linear subarrays of microstrip patches. For 4 to 7 subarrays, the best solution seems to be the circular array, or the three-segment array in some cases. For 8 subarrays or higher, the planar array is the most suitable.

2.3.1 Array Factor for Linear Arrays

Linear arrays are formed by placing elements next to each other along an axis.

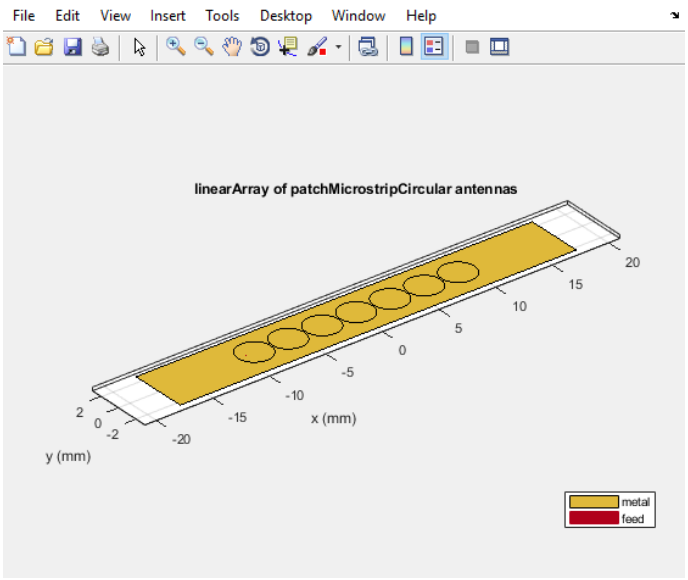

Figure 2: Linear array with circular patch microstrip antennas in Matlab.

The array factor is a function which describes the positions of the antennas and their weights in a specific array, and for linear arrays it is given by

$$AF = 1 + e^{+j(kd\cos\theta+\beta)} + e^{+j2(kd\cos\theta+\beta)} + \cdots + e^{j(N-1)(kd\cos\theta+\beta)}$$

$$AF = \sum_{n=1}^{N} e^{j(n-1)(kd\cos\theta+\beta)} \tag{2-2}$$

The above formula can also be written as

$$AF = \sum_{n=1}^{N} e^{j(n-1)\psi}, where\ \psi = kd\cos\theta + \beta \tag{2-3}$$

where N is the number of elements in the array, k is the wave number, which is equal to $2\pi/\lambda$, d is the spacing between the elements, θ is the angle between the array axis (i.e. x axis in our case, see Figure 3) and the direction of the wave propagation and β represents the phase by which the current in each element leads the current of the preceding element.

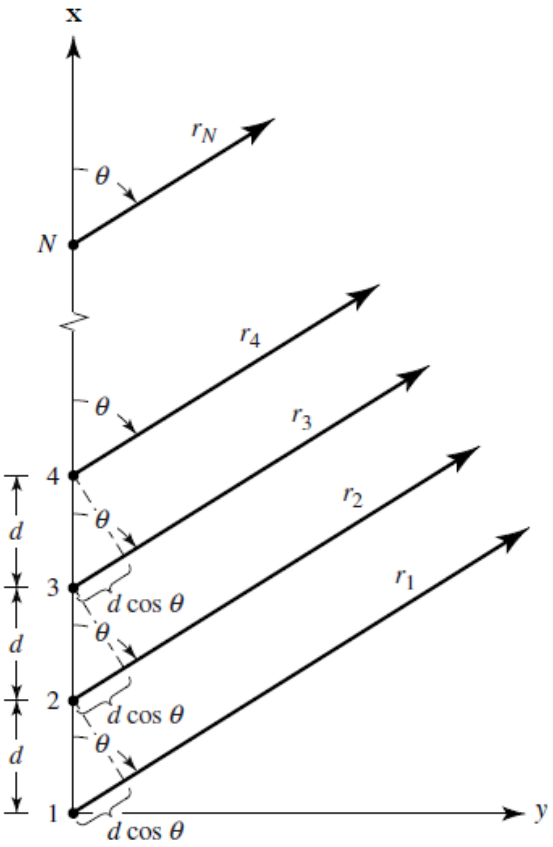

Figure 3: Linear antenna array on x axis and angles of interest [18].

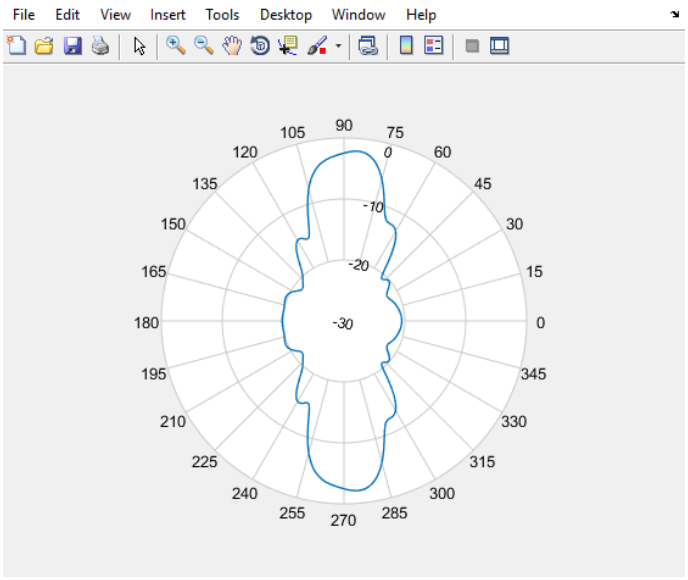

Figure 4: Array factor of the array in Figure 1, for the x-y plane (i.e. the plane containing the array) with uniform amplitude and zero phase in Matlab.

In this thesis, linear antenna arrays are examined using two different synthesis algorithms, which will be described in detail in Chapter 3.

2.3.2 Array Factor for Planar Arrays

Planar arrays are formed by placing elements in both vertical and horizontal positions. They have additional variables than linear arrays to control the array pattern. This allows them to provide a more symmetrical pattern along with lower side lobes.

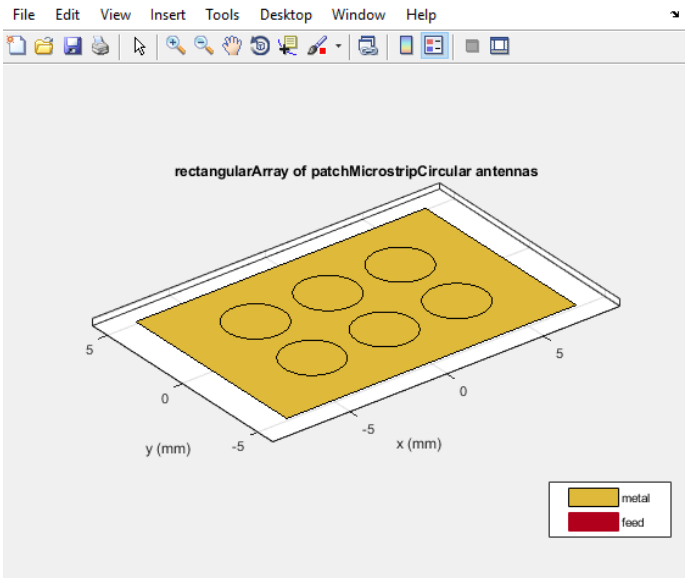

Figure 5: Rectangular array with circular patch microstrip antennas in Matlab.

The array factor for planar arrays is given by

$$AF = \sum_{n=1}^{N} I_{1n} \left[\sum_{m=1}^{M} I_{m1} e^{j(m-1)(kd_x \sin\theta \cos\varphi + \beta_x)} \right] e^{j(n-1)(kd_y \sin\theta \sin\varphi + \beta_y)} \quad (2\text{-}4)$$

where M is the number of elements along the x axis, I_{m1} is the excitation coefficient of each element in x axis and I_{1n} for y axis, k is the wave number, which is equal to $2\pi/\lambda$, d_x is the spacing between the elements on the x axis and d_y in the y axis, θ and φ are the angles of interest and β_x represents the phase by which the current in each element leads the current of the preceding element on the x axis and β_y in the y axis. Equation (2-4) can be simplified for proportional or uniform amplitude excitation coefficients on the x and y axis.

2.3.3 Directivity

The total directivity of an antenna is given by [18]

$$D_0 = D_\theta + D_\varphi \tag{2-5}$$

where D_θ and D_φ are the partial directivities

$$D_\theta = \frac{4\pi U_\theta}{(P_{rad})_\theta + (P_{rad})_\varphi} \tag{2-6}$$

$$D_\varphi = \frac{4\pi U_\varphi}{(P_{rad})_\theta + (P_{rad})_\varphi} \tag{2-7}$$

The radiation intensity U in a given direction is defined as the power radiated from an antenna per unit solid angle and is calculated by the following formula

$$U = r^2 W_{rad} \tag{2-8}$$

where r is the distance and W_{rad} is the radiation density.

The total radiated power is given by

$$P_{rad} = \int_0^{2\pi} \int_0^{\pi} U \sin\theta \, d\theta \, d\varphi \tag{2-9}$$

In the Matlab simulation the radiation pattern is calculated with the help of the array factor, thus without using (2-8) and radiation density, a method which is examined in paragraph 3.2.

2.4 Beamforming

Sharing the same medium, which affects the quality the users are experiencing, path loss and Light Of Sight (LOS) issues can be dealt with the deployment of beamforming antennas. Beamforming gains are also able to improve the link quality for devices at the cell edges. In more detail, by controlling the amplitude and phase of each antenna array element the electromagnetic radiation is manipulated and steered to the designated direction. In addition to improving the signal-to-noise ratio (SNR), beamforming also directs the signal to other directions, thus reducing interference. The transmissions are now separated in the spatial domain leaving no restrictions for the use of time and frequency resources. This multiple access technique is known as Spatial Division Multiple Access (SDMA). Millimeter waves, due to their nature, can take advantage of SDMA, which can enhance the communication system further by boosting network's coverage,

improve signal quality by mitigating interference from nearby users and multipath, and increase system's capacity. The directivity and the steering capabilities depend on the type of the antenna array. For example, circular arrays are able to larger angles than linear arrays, but the later have better directivity. Furthermore, by using a wider beamwidth antenna the received signal is more accurate, while a narrow beamwidth has better directivity [1].

Three popular types of beamforming are analog, digital and hybrid beamforming. Digital baseband beamforming permits transmission of multiple spatial streams of data (MIMO). This leads to higher throughput, but the cost to make it possible is vast per antenna due to size, weight and power consumption. Analog baseband beamforming is cost and energy efficient, but only one stream of data can be transmitted. Despite this fact, it seems to be the dominant architecture for millimeter wave communications due to its simplicity [19]. The combination of both digital and analog beamforming architectures, known as hybrid beamforming, provides narrow beams with phase shifters at the analog domain and combines them with the flexibility of digital domain. In [20], authors suggest a hybrid beamforming architecture where the directive beams are formed with analog phase shifters, to overcome the increased path loss of millimeter waves, and digital beamforming is used to enable MIMO. This shows exactly how parts of analog and digital beamforming can be combined to achieve a cost-efficient architecture, maintain tolerable path loss and at the same time take advantage of advanced multi-antenna techniques.

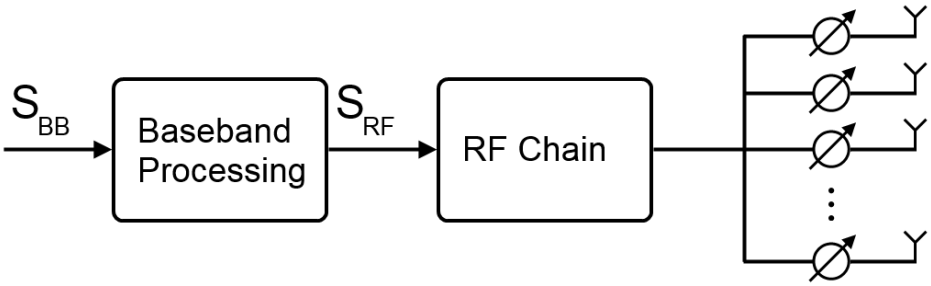

Figure 6: Analog Beamforming Architecture [17].

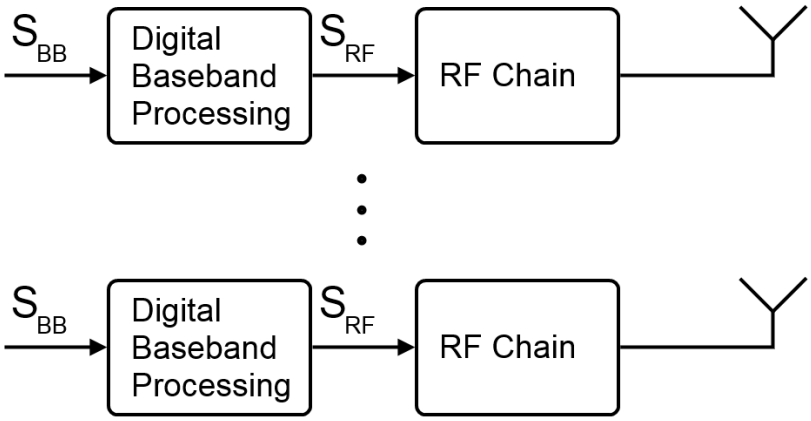

Figure 7: Digital Beamforming Architecture [17].

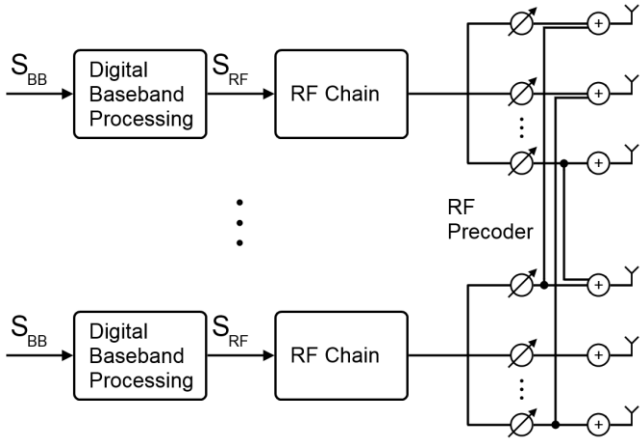

Figure 8: Hybrid Beamforming Architecture [17].

For millimeter wave frequencies the control of the array is implemented with phase shifters in order to maximize power efficiency and reduce the cost of equipment [1]. The authors in [21] conduct measurements at 38 GHz for outdoor cellular channels, providing information for a system able to implement antenna beam steering. Apart from the data that can be examined from the results of the research, it is also an example of how SDMA is capable to increase spectrum reuse and capacity, decreasing interference and provide extended range, proving that narrow beams, which are generated by highly directional and steerable beam antenna arrays, produce much less interference. In [19], authors propose a 6-sector configuration at the base station, rather than the typical 3-sector configuration, which will boost throughput and at the same time reduce interference. At the mobile station two sets of patch antennas are examined, due to the fact

that they are easy to implement and are able to provide full coverage and steering potentiality. Ultimately, the results show that by using beamforming on both the transmitter and the receiver, SNR is increased. In addition, there is room for further improvement by implementing technologies such as SDMA, MIMO and spatial multiplexing.

A challenge which rises with the introduction of beamforming is antenna beamforming training protocols. Antennas must initialize a protocol before any communications starts in order to discover the best beam direction to pair efficiently with a device and take advantage of SDMA. Beamforming training is an active field of research, in pursuance of an efficient algorithm.

The authors in [22] propose a beamforming technique called “beam coding”, taking the in-packet beamforming training a step further. Beam coding allows the transmission of packetized data in multiple beams, rather than just one. The transmitter sends the same packet, while the receiver switches between the different beams and then compares the received signal for each beam. Lastly, by using Walsh code and Golay sequences orthogonality is preserved, even in the presence of multipath. All antenna elements are assumed to be omnidirectional and the array is a non-uniformly weighted linear array, although the use of uniformly weighted uniform linear array is also feasible. Simulations are performed using the living room model described in IEEE 802.11ad.

A transmit precoding and receive combining for millimeter wave applications, based on Singular Value Decomposition (SVD) through a multi-stage iterative method for 60 GHz millimeter wave communications systems is examined and proposed in [23], especially for communication systems with only a few RF chains. This technique will enhance the data rates supported by the system.

In the 60 GHz spectrum the number of multipath components is limited, due to the high directionality of the beams. This exposes the signal to channel fluctuations. By changing to a “better” path, the interference from human movement and/or blockage can be mitigated. The aforementioned can be achieved by using beamforming techniques to acquire the different channel state for each path. In [24], Training Beam (TB), Successive Angle of Arrival (AoA) and Matrix Pencil (MP) techniques are used with a 16-antenna Uniform Linear Array (ULA) and two different channel models: IEEE 802.11ad conference room and uniform Angle of Departure (AoD). The results show Successive AoA estimation scheme to be the most accurate.

The authors in [25] use Linear Dynamical System (LDS) model to examine the drops in signal power, in order to determine that the type of errors cannot be designated by only one factor. The access point is stationary and the second-best path is provided a priori for the simulations. For beam misalignment, when the total signal power is dropped beyond a chosen threshold, the access point will send probing packets to correct it. If this cannot increase the signal power to an appropriate level, then it means that there is a displacement error. To correct this, the access point initializes the feedback beamforming training on the neighboring beams. The authors' method shows that errors can be detected and corrected with only a small amount of resources.

In [26], authors study SDMA in 60 GHz focusing on the estimation of the optimal transmit and receive beamforming vectors, and also develop an iterative antenna training protocol suggested for systems with more antenna elements than RF chains.

Due to time and space restrictions for this thesis, an efficient beamforming protocol lies outside the scope of the thesis. However, the radiation patterns produced by the synthesis methods described in Chapter 3 can be, in principal, combined with the aforementioned works to create an adaptive beamforming algorithm.

2.5 How the previous concepts combine together in 5G

As mentioned, 5G is going to introduce new applications with a diverse set of requirements. In order to meet those requirements for each application, 5G networks will need to very flexible with respect to resource allocation and QoS. SDN separates the control plane from the data plane and allows the network to be "programmable" according to a set of user-defined conditions and events, thus providing the necessary flexibility. C-RAN, by gathering all BBUs in a single location, allows for easier (and cost-effective) signal processing which may benefit multiple base stations, as the information is available at a single place. Thus, the combination of SDN and C-RAN offers high flexibility, but there is still a need for high throughput, extended coverage and lower interference. To address those issues, we employ millimeter waves, which offer a lot of bandwidth in this band. However, millimeter waves suffer from high path loss and that is why the creation of antenna arrays is essential. Antenna arrays can create beams in different directions, thus reducing interference and increasing SNR. Investigating the radiation patterns of antenna arrays and synthesizing suitable patterns will be the topic of this thesis.

3 Methodology

Matlab is used for the simulations, specifically the Antenna Toolbox. The Antenna Toolbox provides a number of functions and applications for the design, analysis and visualization of antenna elements and arrays. In more detail, it features an antenna catalog with pre-built antenna elements which can be parameterized at will, the ability to design or import custom antennas or arrays for further analysis and the ability to design linear, rectangular, circular and conformal antenna arrays, with the option to do it over infinite ground plane in order to speed up the process for large arrays. Impedance, S-parameters and voltage standing wave ratios can be computed using the Method of Moments described in [27]. The electromagnetic field can be computed and visualized in 2D or 3D over the desired planes, while the far-field radiation pattern can be used to design antennas and antenna arrays. Lastly, for more features Antenna Toolbox can be combined with Phased Array System Toolbox and Communications Toolbox to design complete communication systems, such as MIMO.

3.1 Antenna Toolbox Functions

Antenna Toolbox provides a plethora of functions, but in this section only those that are used in the thesis' simulations will be described.

3.1.1 Design

The design function is used to create an antenna of a specific resonance frequency. The syntax of this function is the following:

```
Variable_Name = design (object, frequency)
```

An example would be the following:

```
el = design (patchMicrostripCircular, 60e9)
```

where a circular patch microstrip antenna is created of 60 GHz resonance.

There are many types of antennas that can be used, such as dipole, monopole, spiral, helix etc., but they will not be examined in the thesis.

3.1.2 LinearArray

The linearArray function creates a linear antenna array based on specified properties. The syntax of this function is the following:

Variable_Name = linearArray (‘Property 1’, value 1, ‘Property 2’, value 2, …)

where the properties that can be used are Element, which is the individual antenna type, NumElements, which is the number of elements in the array, ElementSpacing, which is the spacing between each element, AmplitudeTaper, which is the excitation amplitude for elements in the array, PhaseShift, which is the phase shift in degrees for elements in the array, Tilt, which is the angle by which array will be tilted in a degrees and TiltAxis, which is the axis around which the array will be tilted. An example with some of these properties is the following:

la = linearArray (‘Element’, el, ‘NumElements’, 5, ‘ElementSpacing’, 1.2*0.005/2)

where a linear array of 5 elements is created with 1.2*0.005/2 spacing between them.

el is the antenna element designed in section 3.1.1.

There is of course the option to create antenna arrays other than linear, such as rectangular, circular and conformal, but they will not be examined in the thesis.

3.1.3 Pattern

The pattern function can be used to calculate or plot, depending on the syntax, the radiation pattern of an antenna, an array or an element in the array in a polar (default) coordinate system. The plotted quantity can be directivity in dBi, gain in dBi, electric field in V/m, power in Watts or power in dBm. The syntax of this function is the following:

pattern (object, frequency, azimuth, elevation)

which plots the radiation pattern of the antenna or array over the specified frequency value for the given azimuth and elevation angles. For example:

pattern (la, 60e9, 0:1:360, 0)

where la is an already defined antenna array (say, a linear array made from three circular patch microstrip elements), 60 GHz is the carrier frequency and the azimuth angles are 0 to 360 degrees, with a step of 1 degree, while the elevation angle is 0. This means the result will be a 2D radiation pattern and can be seen below.

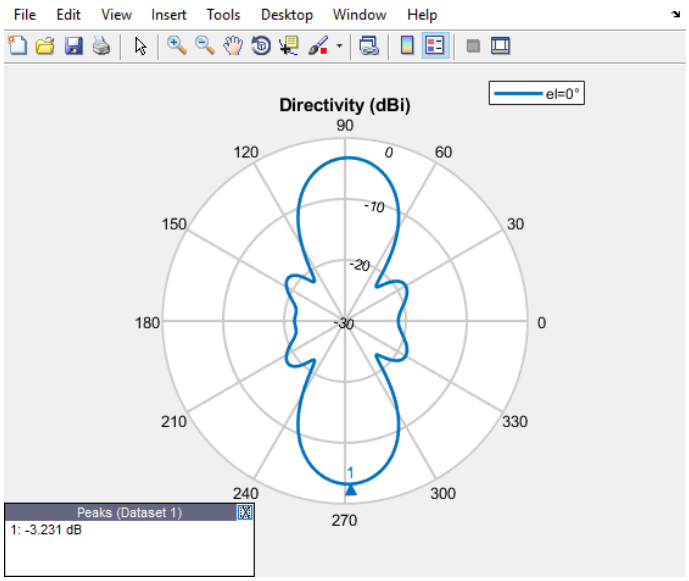

Figure 9: 2D Radiation pattern plotted from pattern (la, 60e9, 0:1:360, 0).

An example of calculation without plotting for pattern is the following:

[DD, azi, elv] = pattern (la, 60e9, 0:1:360, 0)

where the directivity results are stored in the matrix DD and the angles in the variables azi and elv.

3.1.4 ArrayFactor

The arrayFactor function is used to calculate or plot the array factor in dB. The syntax of this function is the following:

arrayFactor (object, frequency, azimuth, elevation)

which plots the array factor of the antenna array over the specified frequency value for the given azimuth and elevation angles. For example:

arrayFactor (la, 60e90, 0:1:360, 0:90)

where la is an already defined antenna array (say, a linear array made from three circular patch microstrip elements), 60 GHz is the carrier frequency and the azimuth angles are 0 to 360 degrees, while the elevation angles are 0 to 90 degrees. This means the result will be a 3D representation of the array factor and can be seen below.

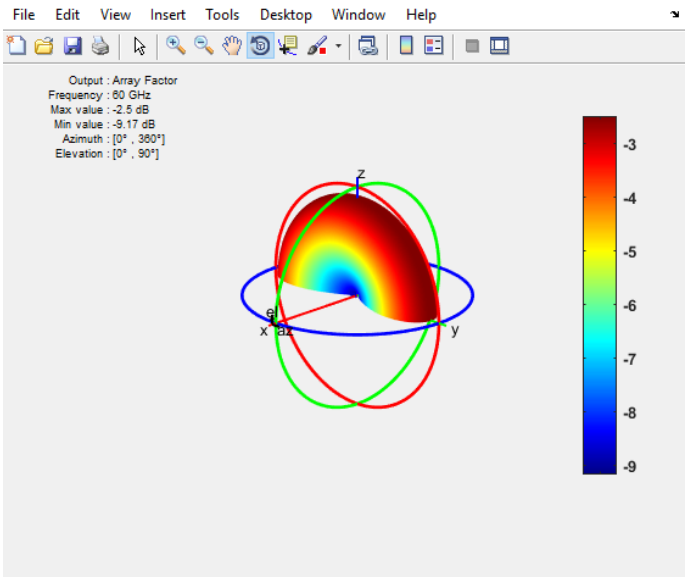

Figure 10: 3D representation of the array factor plotted from arrayFactor (la, 60e9, 0:1:360, 0:90).

An example of calculation without plotting for arrayFactor is the following:

[AFR, azi, elv] = arrayFactor (la, 60e9, 0:1:360, 0:90)

where the directivity results are stored in the matrix AFR and the angles in the variables azi and elv.

3.1.5 Polarpattern

The polarpattern function is able to create a polar plot based on amplitude values and angles in degrees. The dimensions of amplitude and angle vectors must match. The syntax of this function is the following:

polarpattern (A, M)

where A is the angles vector and M is the amplitudes vector. For example, let us say that we have already used [DD, azi, elv] = pattern (la, 60e9, 0:1:360, 0), so DD is the amplitude vector and 0 to 360 degrees is the angle vector. So:

polarpattern (0:1:360, DD)

produces the following figure.

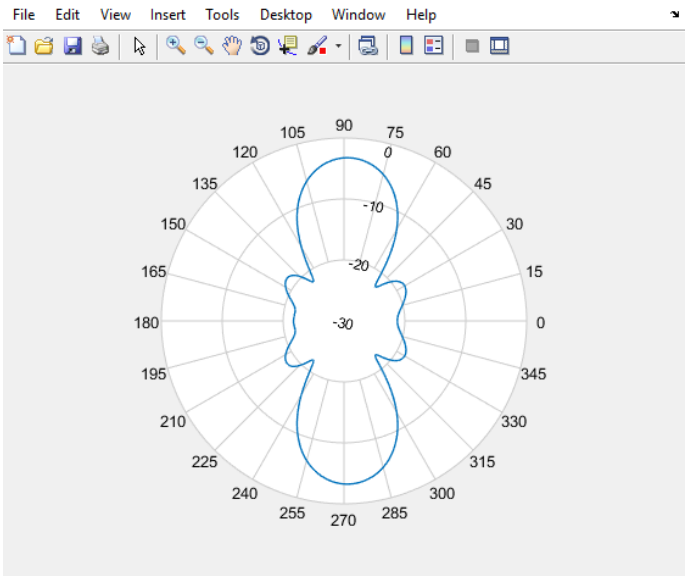

Figure 11: Pattern in polar coordinate system plotted from polarpattern (0:1:360, DD).

3.2 Array Factor Trick

During the Matlab simulations a “trick” was used in order to complete the simulations faster without the need of enormous computational resources. The total radiation pattern of an antenna array can be computed by the multiplication of the radiation pattern of a single element of the array and the array factor of the antenna array. This method overcame the extensive amount of RAM Matlab needs to compute the total radiation pattern of a large antenna array. To make it more clear, here is an example:

1. Creating an antenna.

   el = design (patchMicrostripCircular, 60e9);

2. Creating an array.

   la = linearArray ('Element', el, 'NumElements', 5, 'ElementSpacing’, 1.2*0.005/2);

3. Calculating the radiation pattern of the circular patch microstrip antenna.

   [Ei, azii, elvi] = pattern (el, 60e9, 0:1:360, 0, 'Type', 'efield');

4. Calculating the array factor of the linear array.

   AFR_dB = arrayFactor (la, 60e9, 0:1:360, 0);

5. Converting the array factor to linear scale.

   AFR = 10.^(AFR_dB/10);

6. Calculating the radiation pattern of the array based on the equation (3-1) below.

   Etotalr = AFR.*Ei;

7. Calculating the radiation intensity of the array.

   Intensity = Etotalr.^2;

8. Calculate the total radiated power of the array based on the equation (2-9) by numerically evaluating the corresponding integral.

9. Use the computed Intensity to find the array directivity, which is in linear scale.

   D= 4*pi * Intensity/Prad

The mathematical expression for the multiplication is based on the equation (2-4) and is given by

$$E_{total}(\varphi,\theta) = AF(\varphi,\theta) \cdot E_{ind}(\varphi,\theta) \tag{3-1}$$

The difference between [18] and Matlab is the angle conventions for the elevation. Specifically, [18] defines the elevation angle θ from the north pole while Matlab measures the elevation angle ν from the equation. Hence, it holds $\nu = 90 - \theta \Leftrightarrow \theta = 90 - \nu$.

3.3 Pattern Synthesis Algorithms

Two separate pattern synthesis algorithms, a linear system-based method and the Fourier transform method, were used in the Maltab simulations, one of which (namely, the Fourier method) produced better resulsts which were closer to the desired patterns than the other. Both algorithms were implemented to produce radiation patterns for the x-y plane (i.e., azimuth angle from 0 to 360 degrees and elevation angle 0 degrees), and the x-z plane (i.e., elevation angle from 0 to 90 degrees for azimuth angle equal to 0 degrees and again elevation angle from 0 to 90 degrees for azimuth angle equal to 180 degrees).

3.3.1 Linear System Method

In this method the excitation coefficients are determined by solving a linear system based on the desired array factor and the array factor calculated by (2-2). The (2-2) formula will be the following, in order to satisfy the angle conventions used by Matlab

$$AF_{(\varphi,v)} = \sum_{m=1}^{M} I_m e^{j(m-1)(kd_x \cos v \cos \varphi)} \quad (3\text{-}2)$$

Notice that any desired radiation pattern of an antenna array can be converted into an equivalent desired array factor $AF_{Desired}$ (see step 6 in section 3.2 above), once the radiation pattern of the individual radiating element is known. Also, since equation (2-2) is linear in terms of the excitation currents I_m, we can compute $AF_{(\varphi,v)}$ for specific values of (φ,v) and **require** the computed array factor $AF_{(\varphi,v)}$ to be equal to the desired array factor $AF_{Desired}$ **at these (φ,v)** values. This leads to a linear system of the form C * I_m = $AF_{Desired}$, where * denotes matrix multiplication and C is a matrix who elements are the coefficients of I_m in equation (3-2). Hence, solving the matrix equations C * I_m = $AF_{Desired}$ results in the excitation coefficients, which will be used in order to calculate a recreated array factor. I_m will be a vector of complex numbers, so their amplitude will be used as an input in the antenna's AmplitudeTaper and their phase as an input in the antenna's PhaseShift. Due to the dimensions of $AF_{Desired}$ and $AF_{(\varphi,v)}$, the number of equations of the system (i.e. the number of angles at which the desired pattern will be sampled) exceeds the number of unknown currents (which is equal to the number of radiating elements in the array). Hence, Matlab will return a least-error-norm solution to the above system, which will be an approximate solution only.

3.3.2 Fourier Transform Method

The excitation coefficients of an antenna array can also be calculated using the Fourier Transform method. The crucial fact supporting the Fourier transform method is that the array factor in equation (3-2) is a periodic function (with period 2π) of the azimuth and elevation angles (φ,v) and I_m then become the coefficients of the Fourier series expansion of the periodic array factor.

The formula for an odd number of array elements (N = 2M + 1) is the following

$$a_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} AF_{(\psi)} e^{-jm\psi} d_\psi \quad (3\text{-}3)$$

where a_m are the excitation coefficients, $AF_{(\psi)}$ is the desired array factor, ψ = kdcosvcosφ and -M ≤ m ≤ M (note: m only takes integer values in this range).

The formula for even number of elements (N = 2M) is the following

$$a_m = \begin{cases} \frac{1}{2\pi}\int_{-\pi}^{\pi} AF_{(\psi)} e^{-j[(2m+1)/2]\psi} d_\psi \,, where\ -M \le m \le -1 \\ \frac{1}{2\pi}\int_{-\pi}^{\pi} AF_{(\psi)} e^{-j[(2m-1)/2]\psi} d_\psi \,, where\ 1 \le m \le M \end{cases} \quad (3\text{-}4)$$

4 Results

In this chapter, a comparison between the two pattern synthesis algorithms described in the previous Chapter will be made to determine which algorithm produces more accurate results. Then, the superior method will be evaluated with a variable number of elements and lastly, a comparison with the same number of elements but with different beam shapes (and beamwidths) will be performed.

4.1 Comparison of Pattern Synthesis Algorithms

4.1.1 The x-y plane

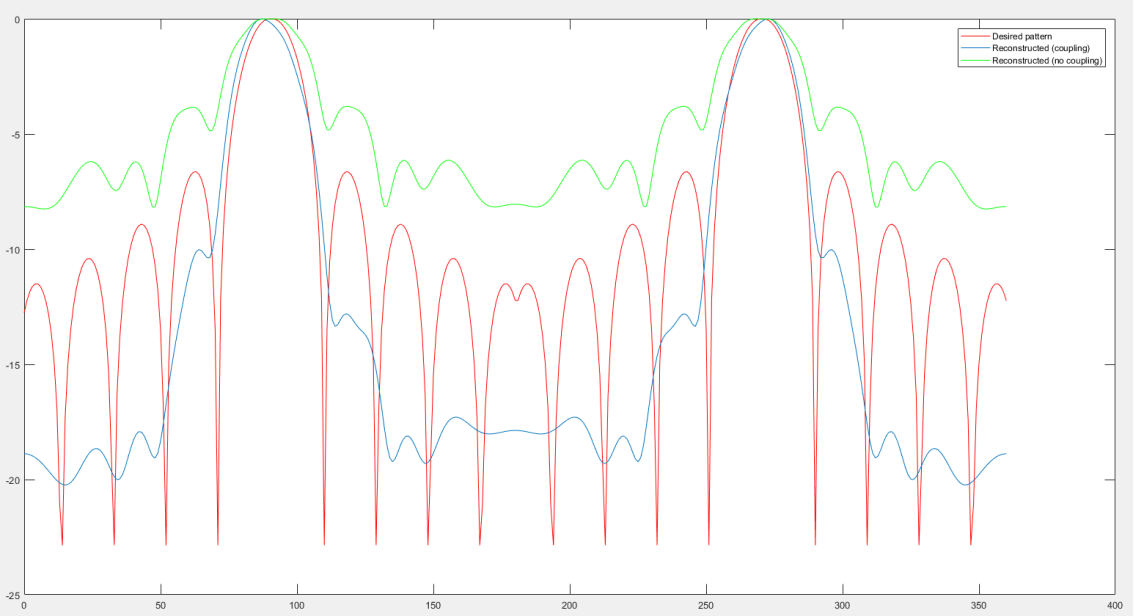

Figure 12: Linear system solution for 11 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

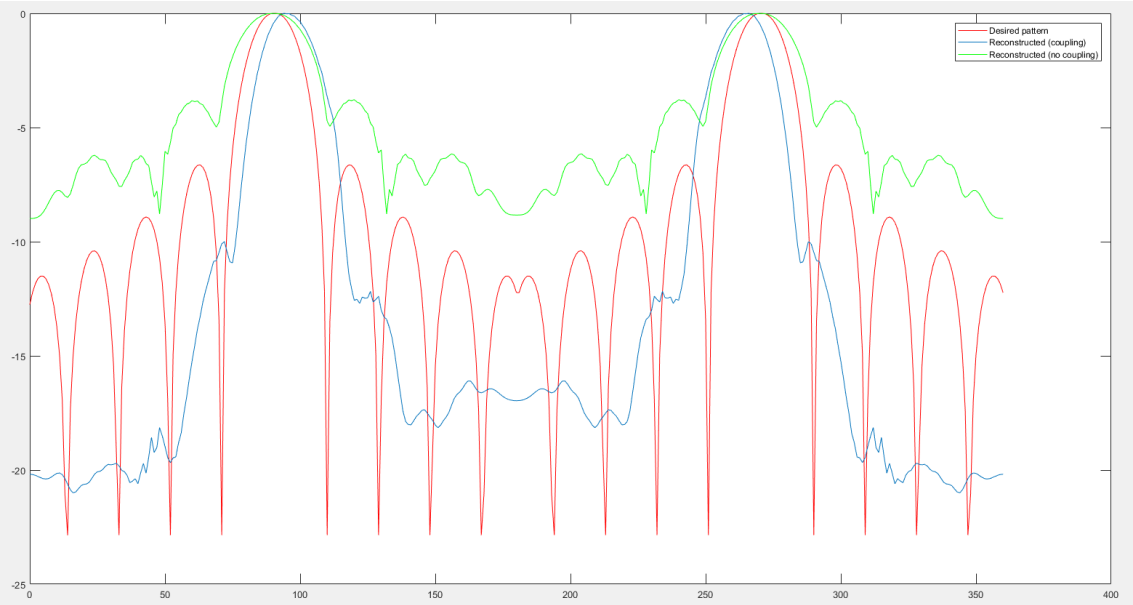

Figure 13: Linear system solution for 51 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

By increasing the number of elements, the expectation was for the reconstructed patterns to be much closer to the desired pattern, a fact that does not happen with the linear system algorithm. By comparing Figures 12 and 13, we observe only a small difference between them, which is not ideal, especially with the addition of 40 more elements.

Note that there are two different solutions in the Figures, namely "coupling" and "no coupling". The difference between them is that "coupling", by definition, also accounts (by modeling the entire array geometry in a mesh structure) for the transfer of energy that occurs between adjacent elements of the array due to their physical proximity whereas the "no coupling" solution ignores this effect. The comparison of the desired pattern should be mainly done with the no coupling solution, as the methodology described in Chapter 3, also does not account for any coupling effects. In addition, when the desired pattern has values below -20 dB the reconstructed pattern is not required to match them as they are considered too low for most practical purposes.

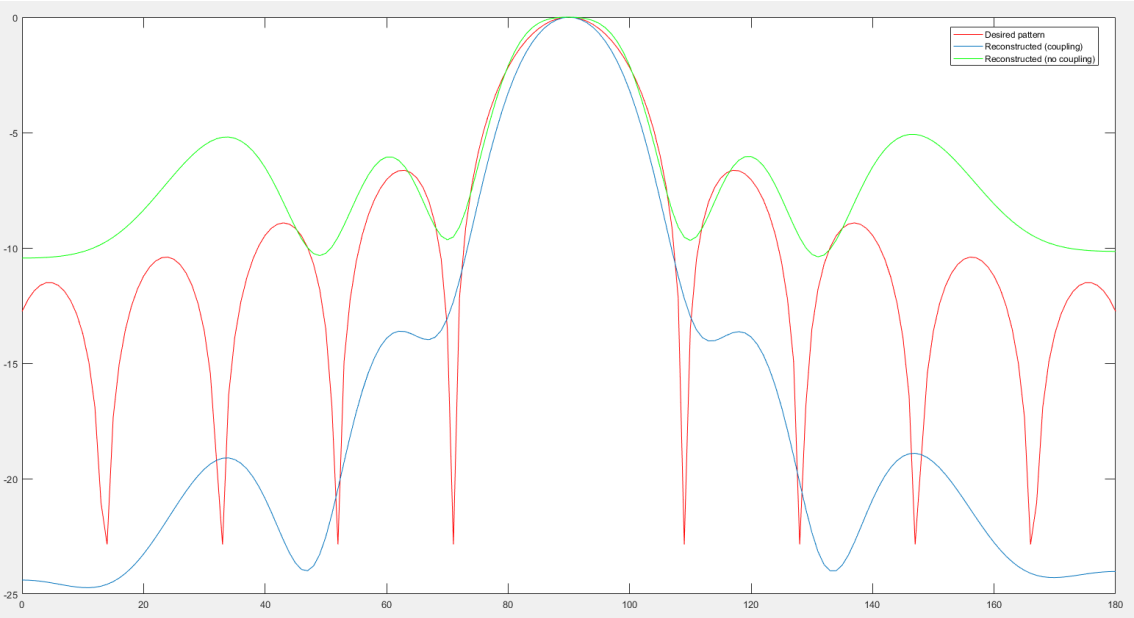

Figure 14: Fourier Transform solution for 11 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

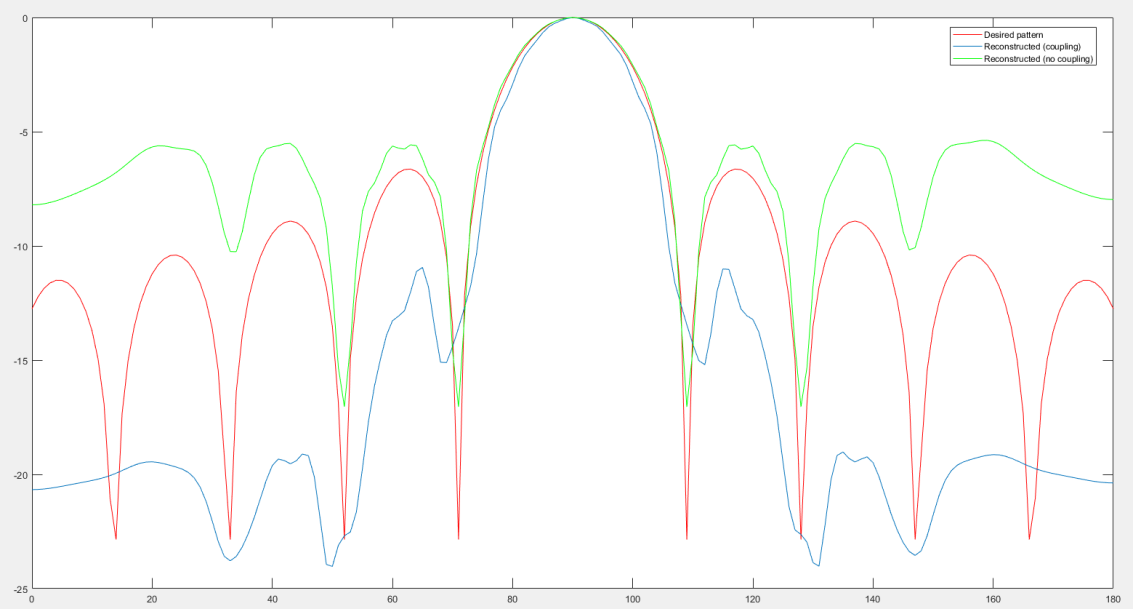

Figure 15: Fourier Transform solution for 51 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

The algorithm based on the Fourier method shows improved results, as the reconstructed patterns are approaching the desired pattern in a better level than the linear system algorithm did, observed by comparing Figure 12 to 14 and Figure 13 to 15. In addition, by increasing the number of elements from 11 to 51, Figures 14 and 15, there is a signif-

icant difference compared to the linear system algorithm, which did not show any remarkable divergence. This particular discrepancy will also be noticed on the x-z plane to a much greater extent.

4.1.2 The x-z plane

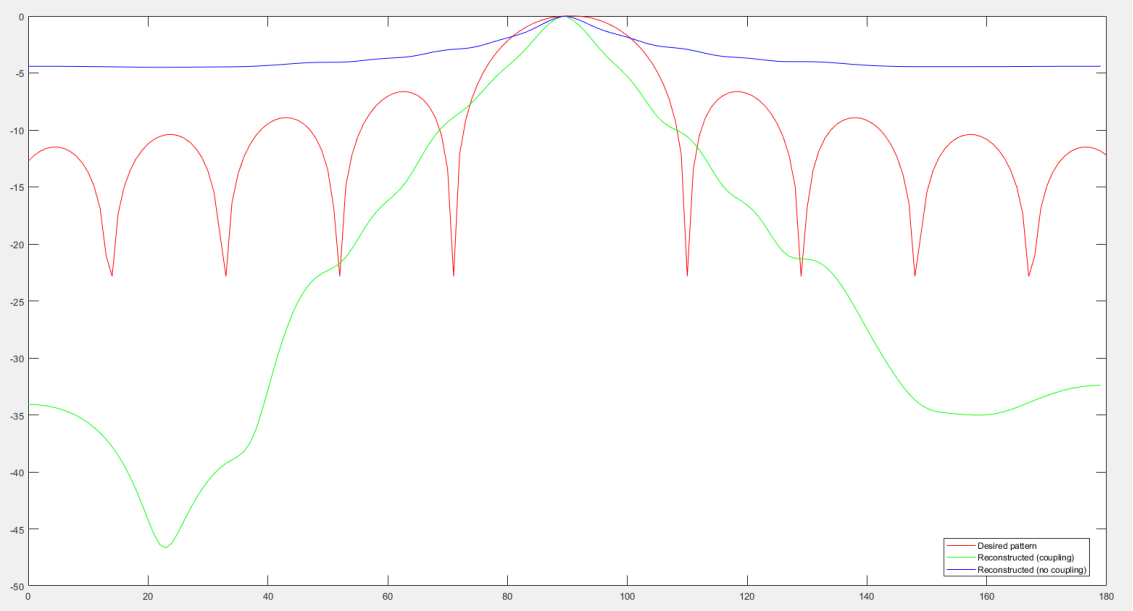

Figure 16: Linear system solution for 11 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

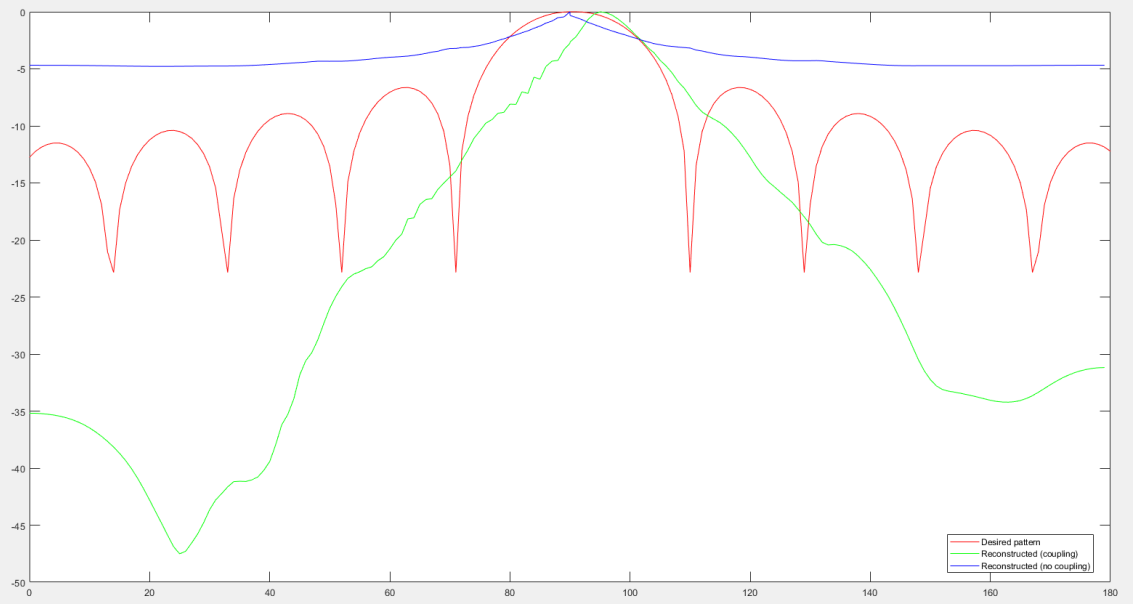

Figure 17: Linear system solution for 51 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

Figures 16 and 17 show that for the x-z plane the linear system algorithm is totally off the target pattern, even for a high number of elements. During the development of the Matlab codes, this was the point where there was definitely a need for at least one more pattern synthesis algorithm, which can accomplish better results in both planes of interest.

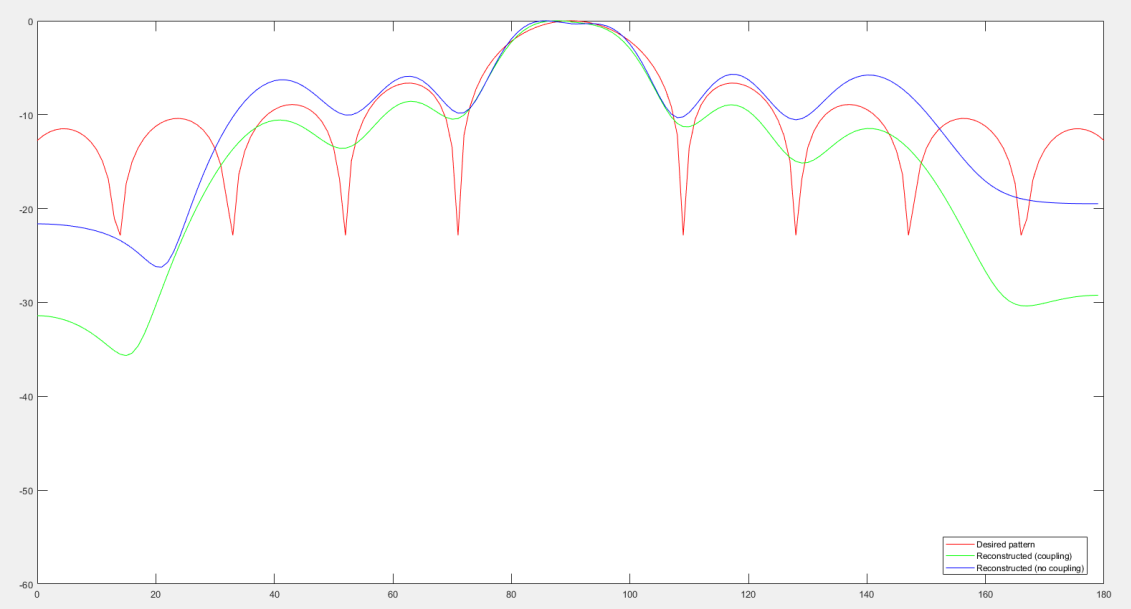

Figure 18: Fourier Transform solution for 11 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

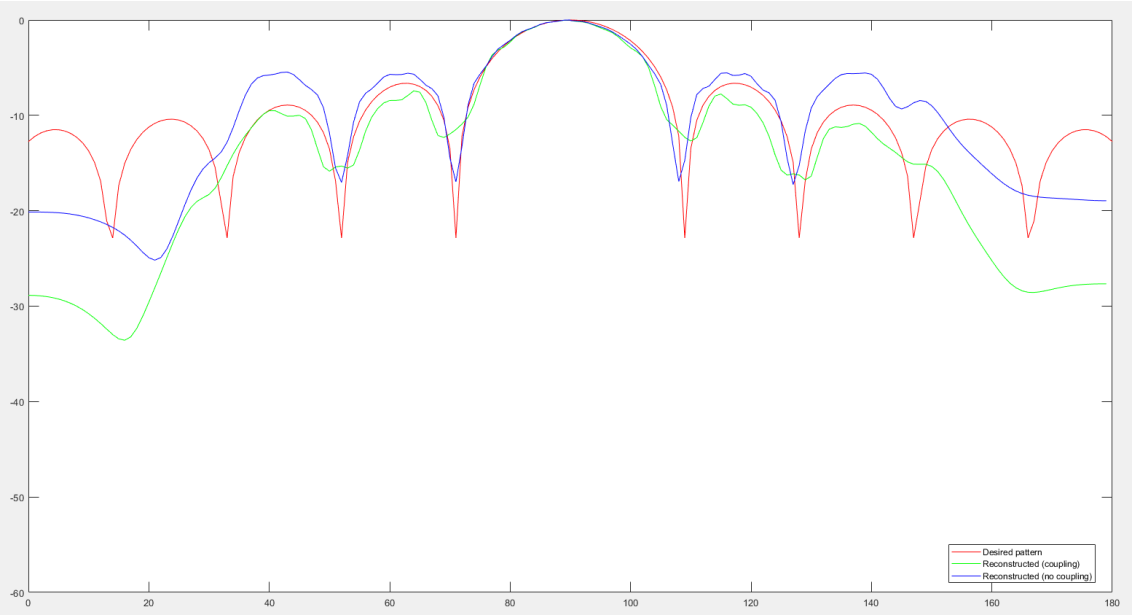

Figure 19: Fourier Transform solution for 51 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

On the other hand, the Fourier transform algorithm shows results close to the desired for 11 elements in Figure 18. In Figure 19, for 51 elements the reconstructed patterns are nearly the same with the desired one. Because of these measurements and the consistency, the Fourier transform algorithm provides, we will focus on this method to examine some other characteristics, such as differences between various numbers of elements and also trying to decrease the beamwidth, which is a key factor for adaptive communication in 5G wireless networks.

4.2 Further Examination of the Fourier Transform Algorithm

4.2.1 Different number of elements on the x-y plane

The expectation of increasing the number of elements in the antenna array is that the reconstructed patterns should be approaching the desired pattern better as the number of elements increases.

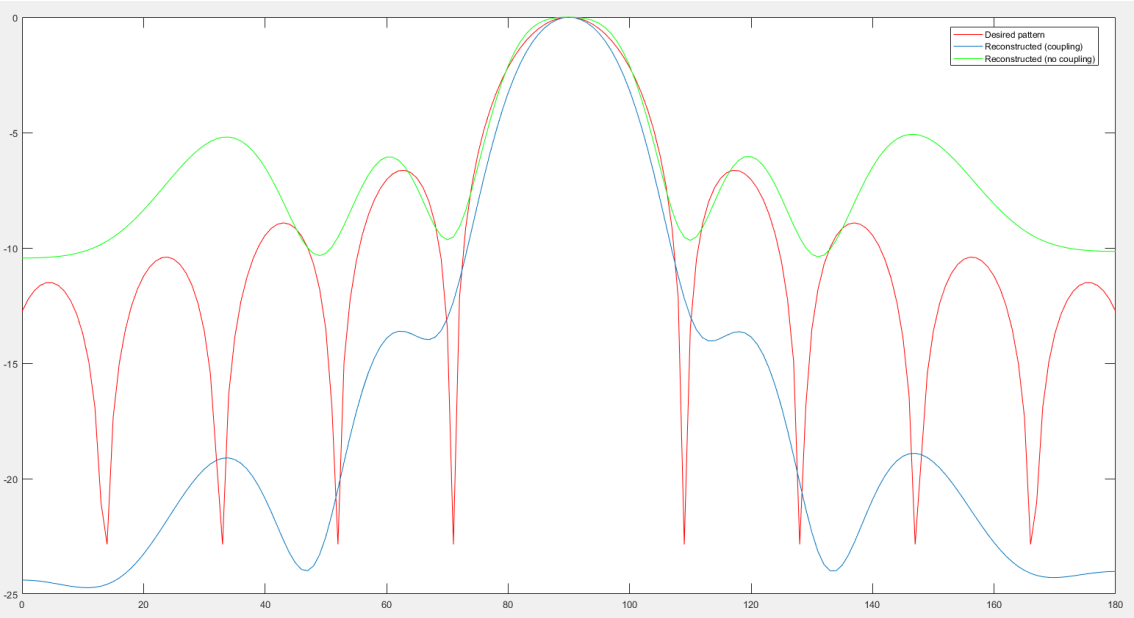

Figure 20: Fourier Transform solution for 11 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

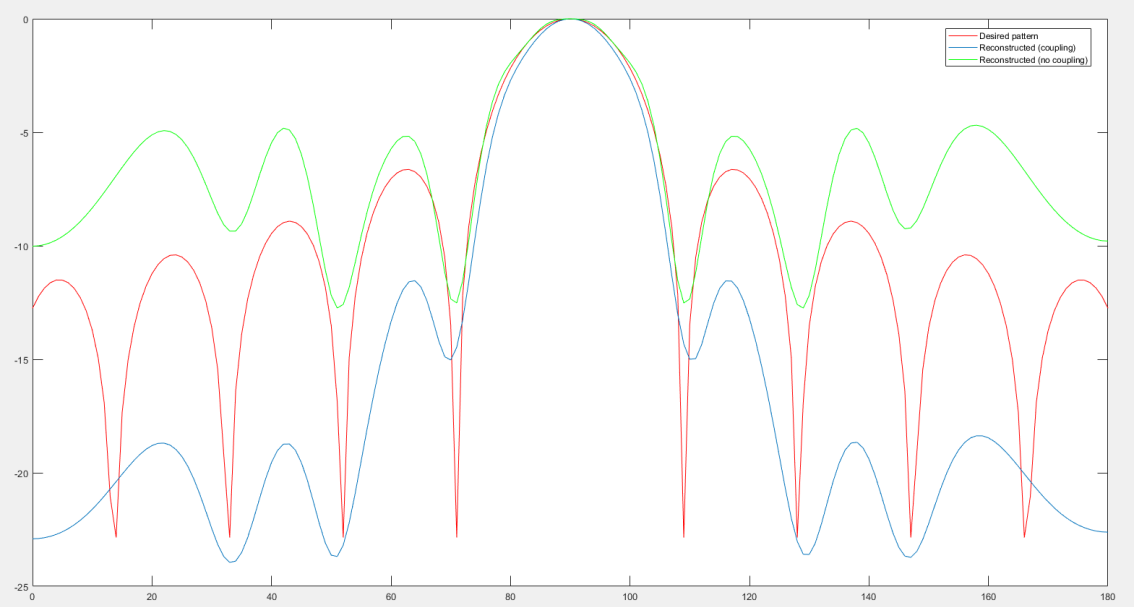

Figure 21: Fourier Transform solution for 21 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

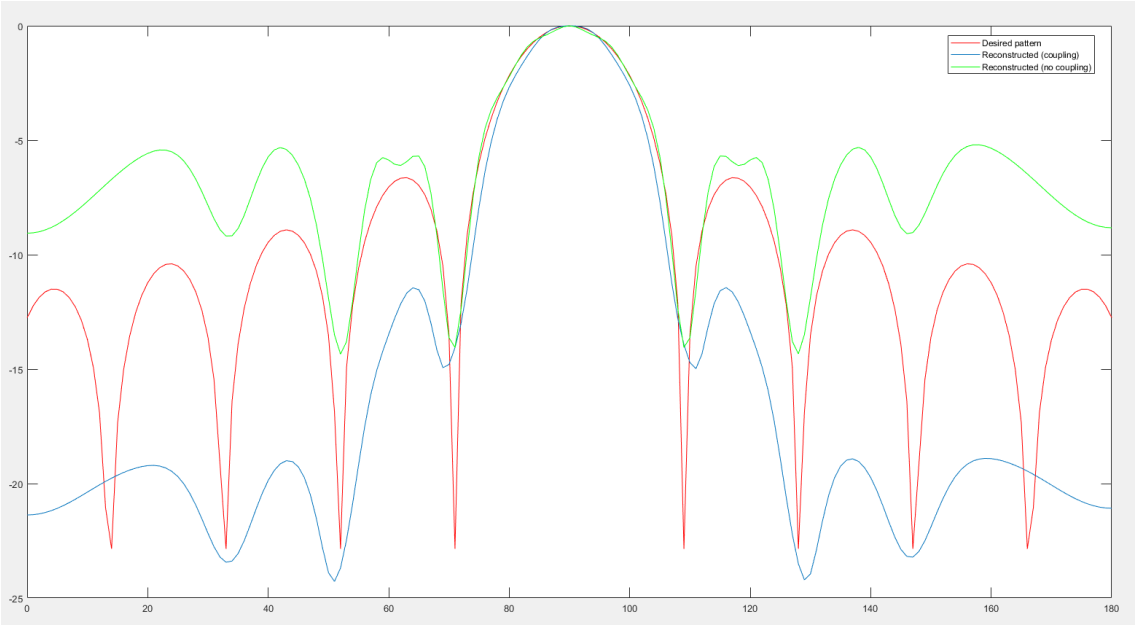

Figure 22: Fourier Transform solution for 31 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

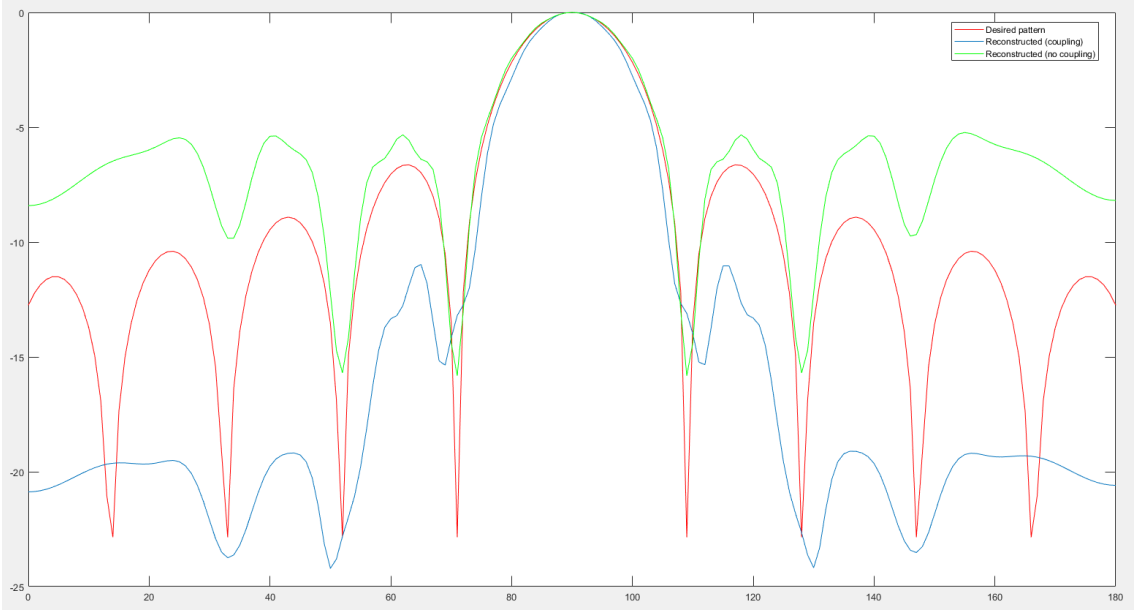

Figure 23: Fourier Transform solution for 41 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

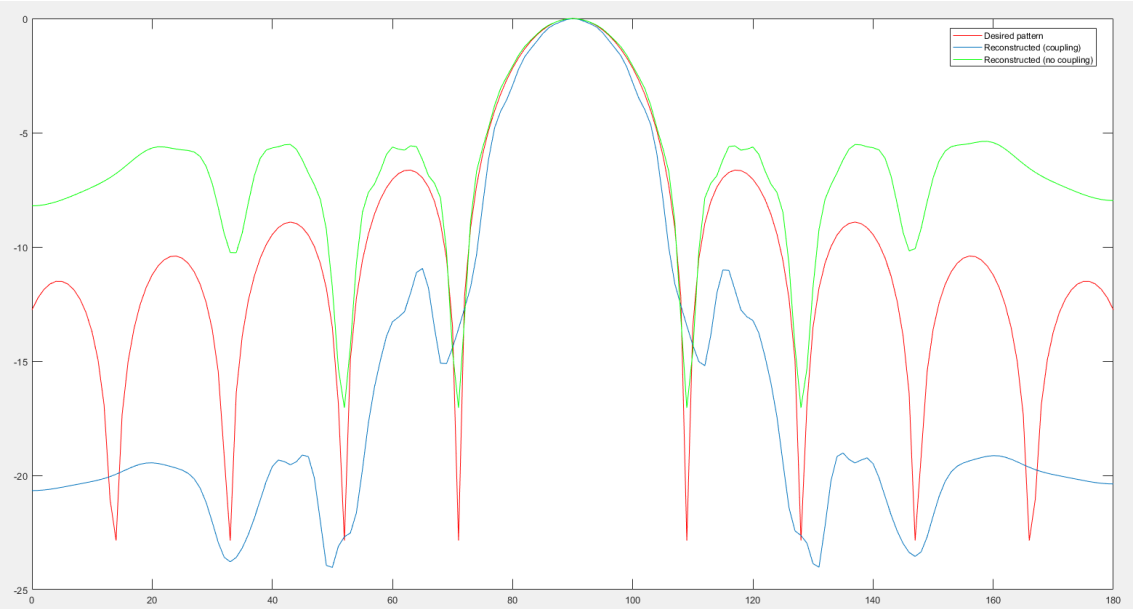

Figure 24: Fourier Transform solution for 51 elements on the x-y plane. Horizontal axis measures azimuth angle in degrees.

Comparison of the Figures 20, 21, 22, 23, 24 shows that indeed the increase of the elements produce better results. For example, the differences between the reconstructed patterns of Figures 20 and 24 are clearly visible, as in Figure 24 the reconstructed patterns approach the desired pattern to a greater degree. Note that in millimeter waves, an array of 51 elements has dimensions of a few cm, which is relatively small.

4.2.2 Different number of elements on the x-z plane

The same trend as in 4.2.1 also appear for the Fourier transform method on the x-z plane.

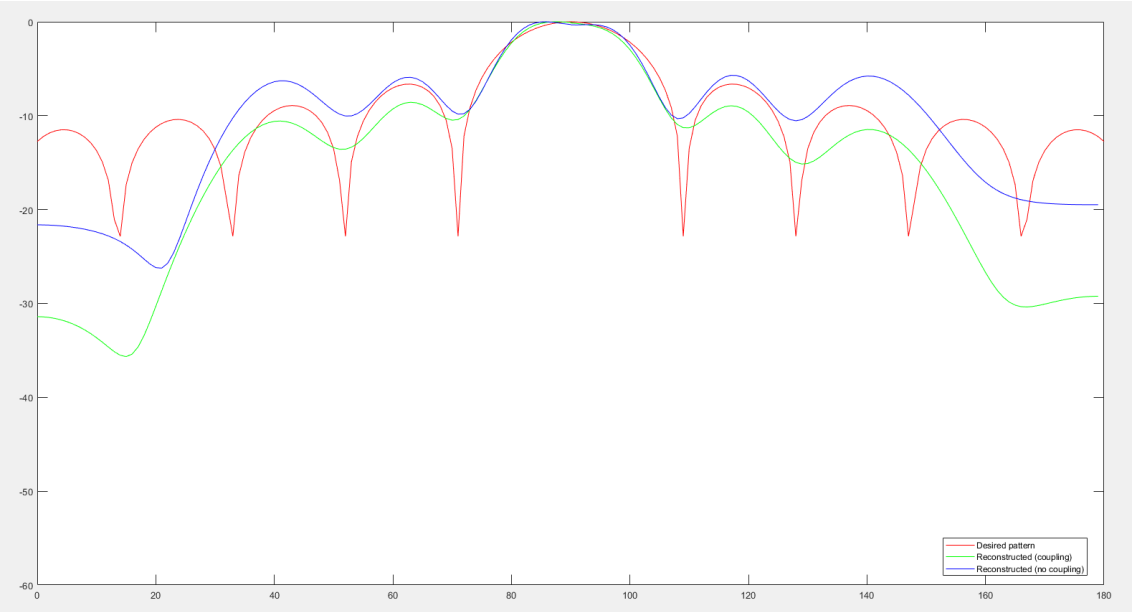

Figure 25: Fourier Transform solution for 11 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

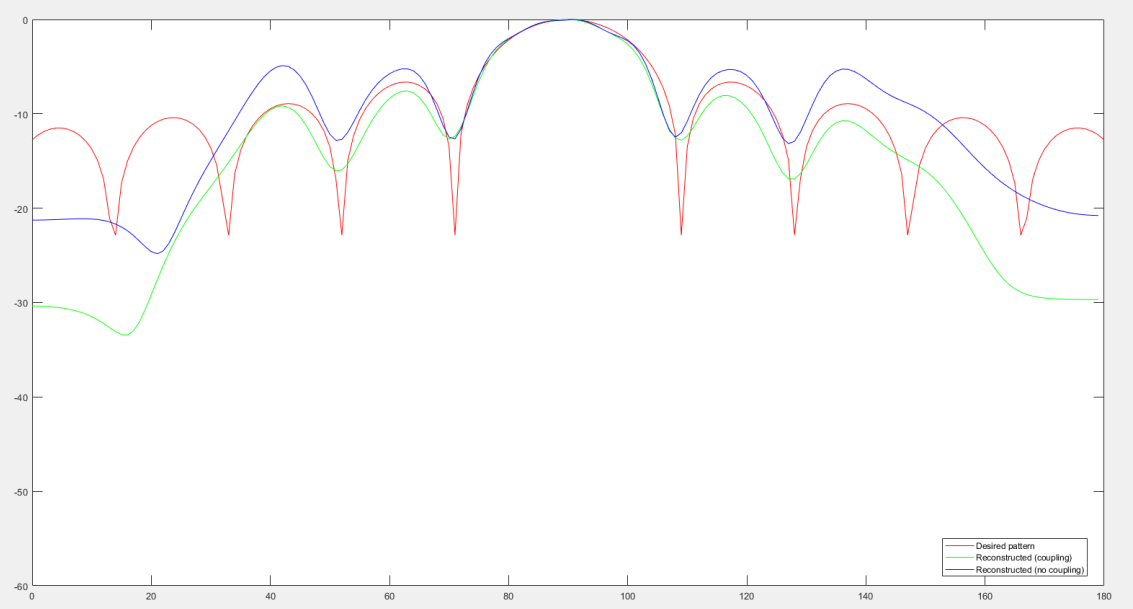

Figure 26: Fourier Transform solution for 21 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

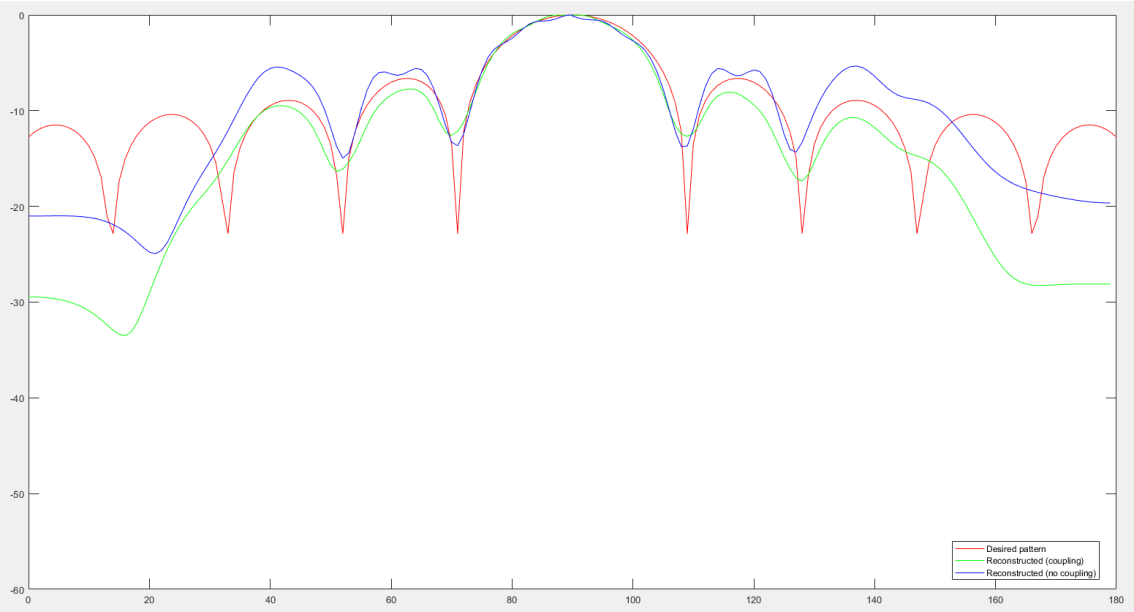

Figure 27: Fourier Transform solution for 31 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

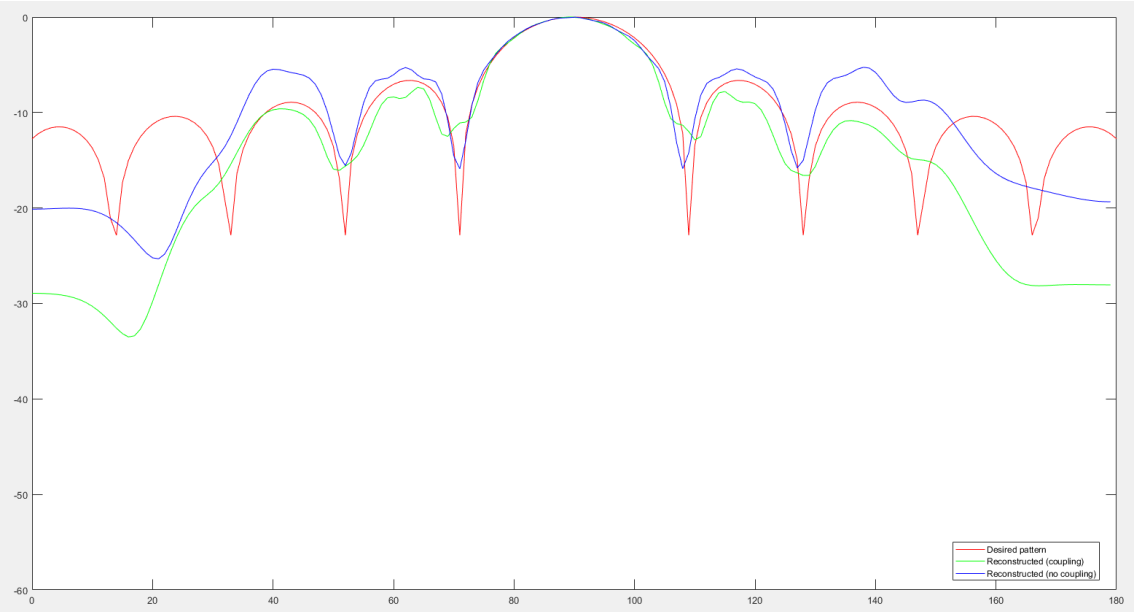

Figure 28: Fourier Transform solution for 41 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

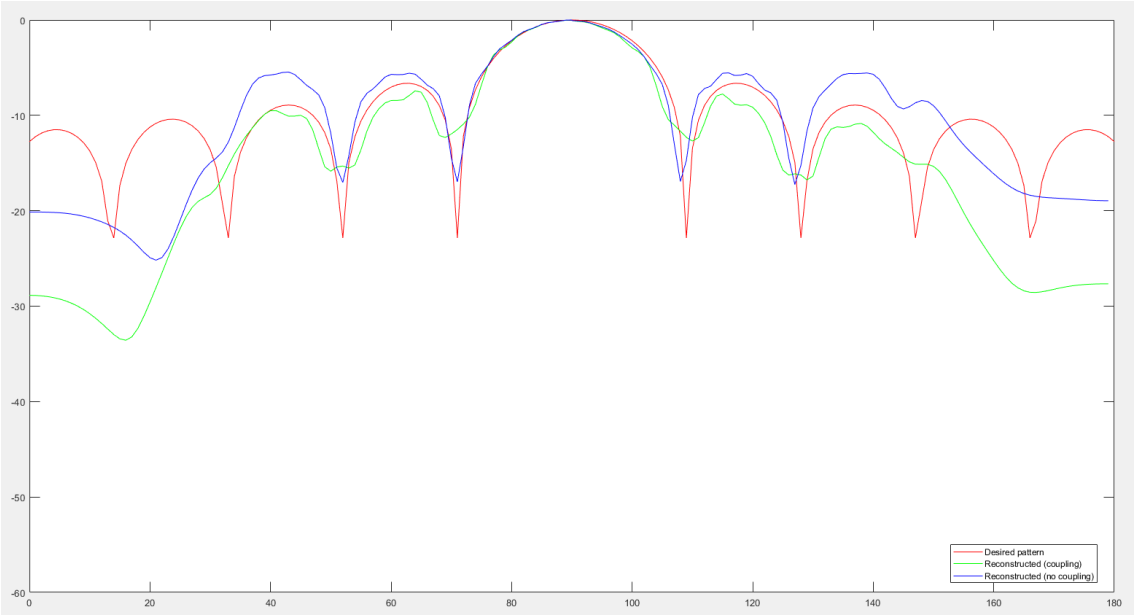

Figure 29: Fourier Transform solution for 51 elements on the x-z plane. Horizontal axis measures elevation angle in degrees.

Similarly to the x-y plane simulations, the same trend is detected on the x-z plane as well. Again, comparing Figures 25, 26, 27, 28, 29 shows that the larger the number of elements the closer the results of the simulation will be to the desired pattern.

4.2.3 Different beamwidth on the x-y plane

Small beamwidth is important for 5G wireless communications in order to compensate for the high path losses of millimeter waves. Hence, it is important to evaluate the ability of any pattern synthesis algorithm (i.e. the Fourier transform method in our case) to produce patterns of variable shapes and widths. The simulations will be performed with a fixed number of elements, which will be 21. The beamwidth through a parameter called *stretch* such that higher *stretch* values lead to lower beamwidths (i.e. more narrow main lobes).

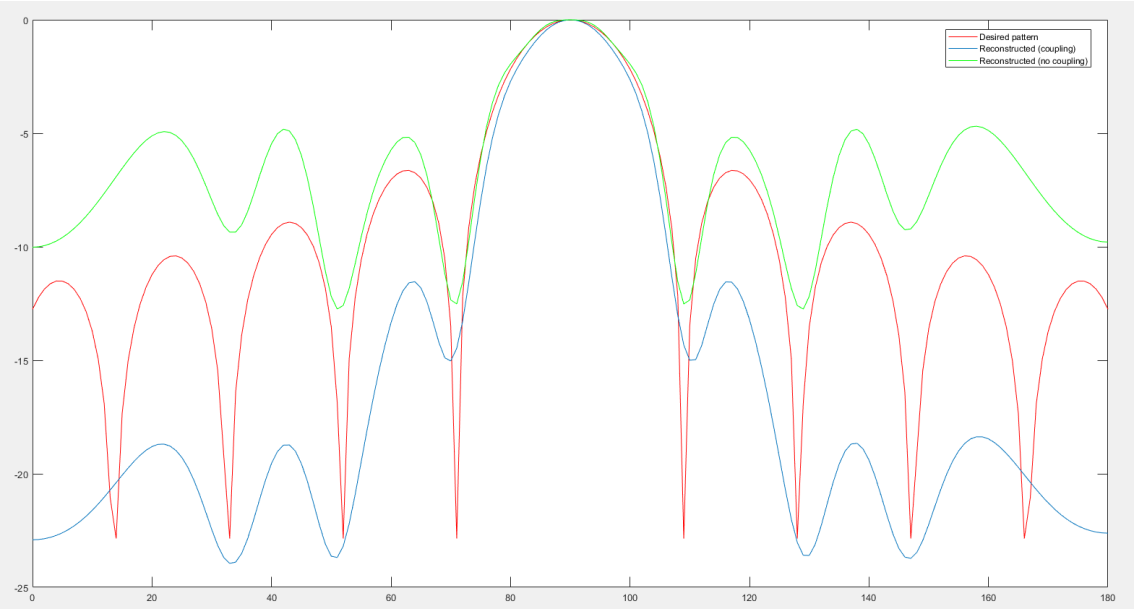

Figure 30: Fourier Transform solution for 21 elements on the x-y plane with a stretch factor of 3. Horizontal axis measures azimuth angle in degrees.

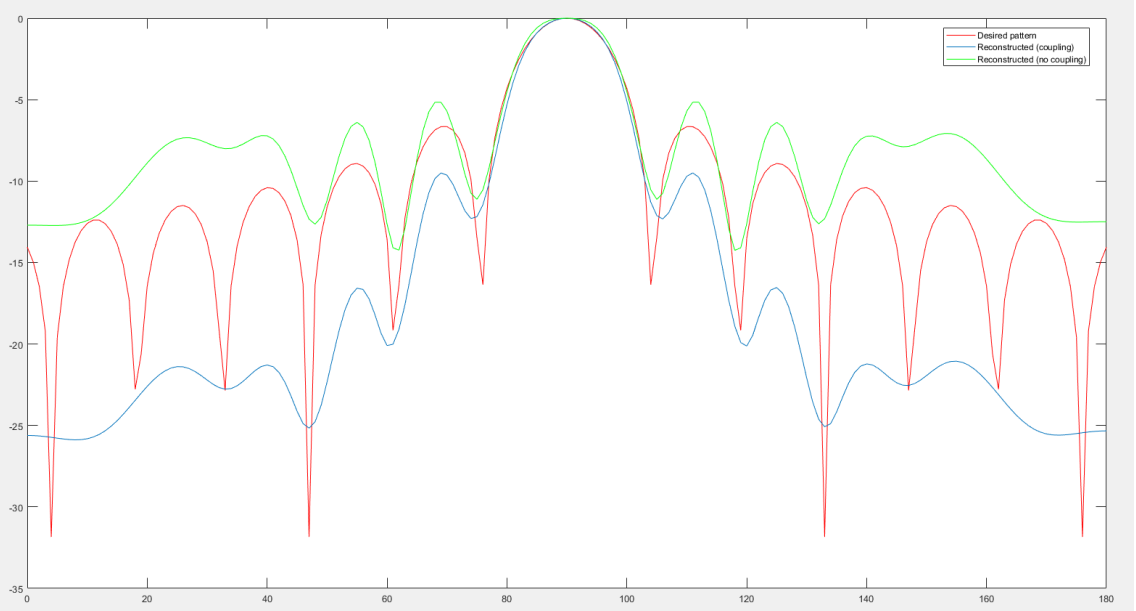

Figure 31: Fourier Transform solution for 21 elements on the x-y plane with a stretch factor of 4. Horizontal axis measures azimuth angle in degrees.

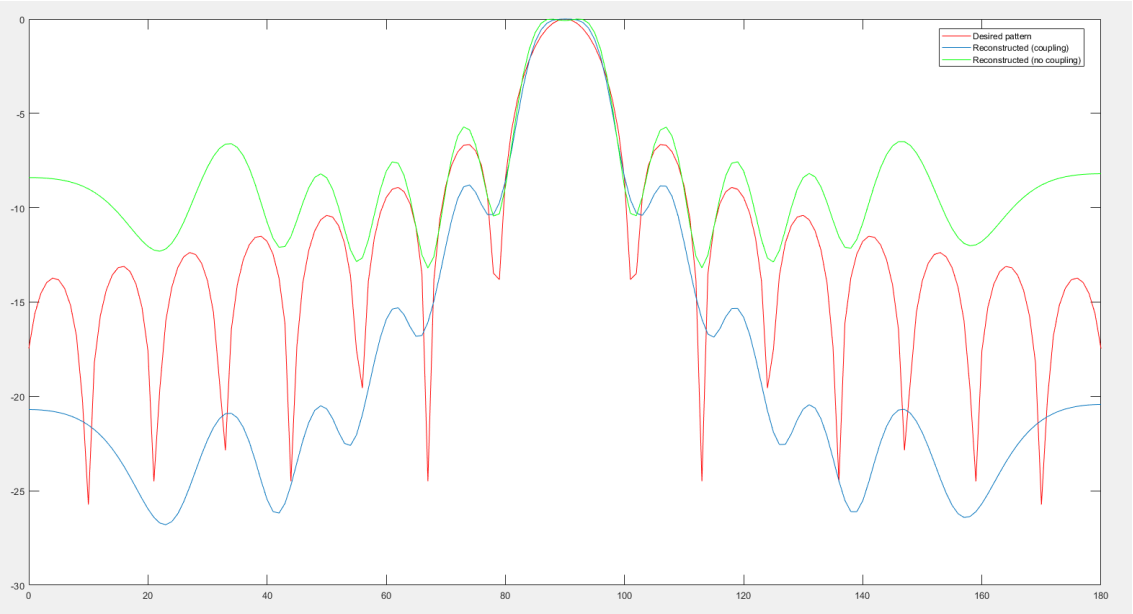

Figure 32: Fourier Transform solution for 21 elements on the x-y plane with a stretch factor of 5. Horizontal axis measures azimuth angle in degrees.

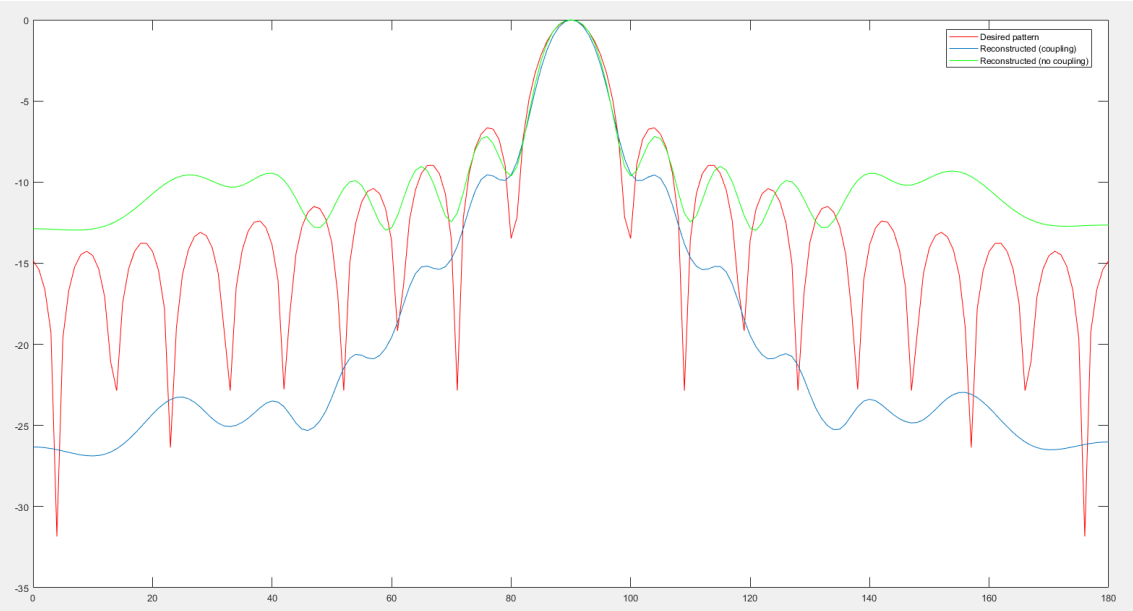

Figure 33: Fourier Transform solution for 21 elements on the x-y plane with a stretch factor of 6. Horizontal axis measures azimuth angle in degrees.

By focusing on the main lobe in Figures 30, 31, 32, 33, a reduction in the beamwidth is observed, as expected before the simulations. For example, in Figure 30, with a stretch factor of 3, the beamwidth is 30 degrees, while the beamwidth in Figure 33, with a

stretch factor of 6, is 16 degrees. The same behavior of results is expected on the x-z plane.

4.2.4 Different beamwidth on the x-z plane

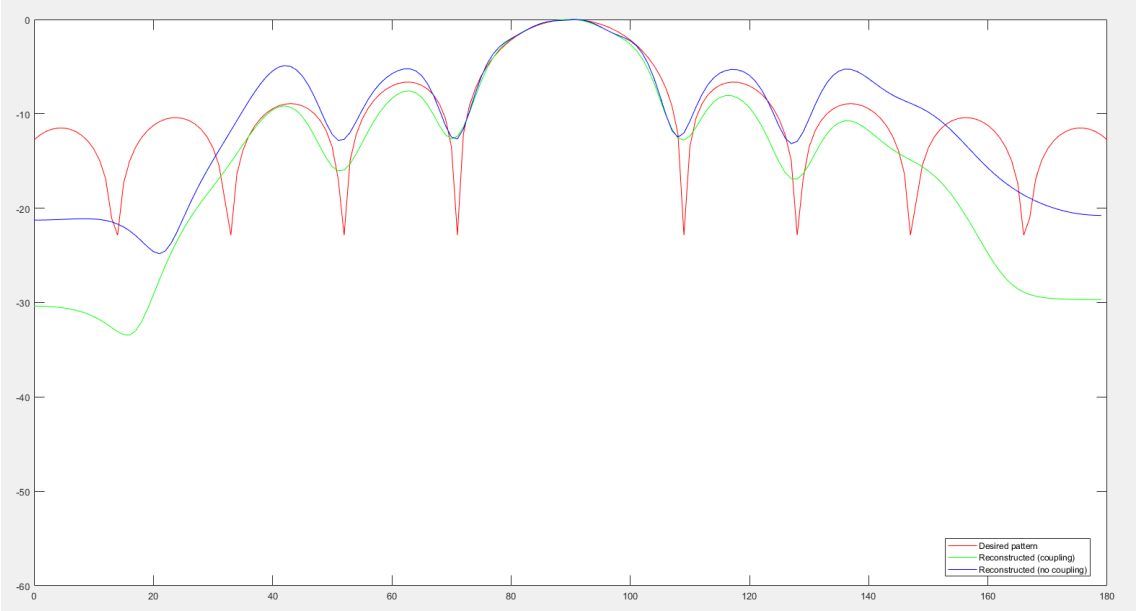

Figure 34: Fourier Transform solution for 21 elements on the x-z plane with a stretch factor of 3. Horizontal axis measures elevation angle in degrees.

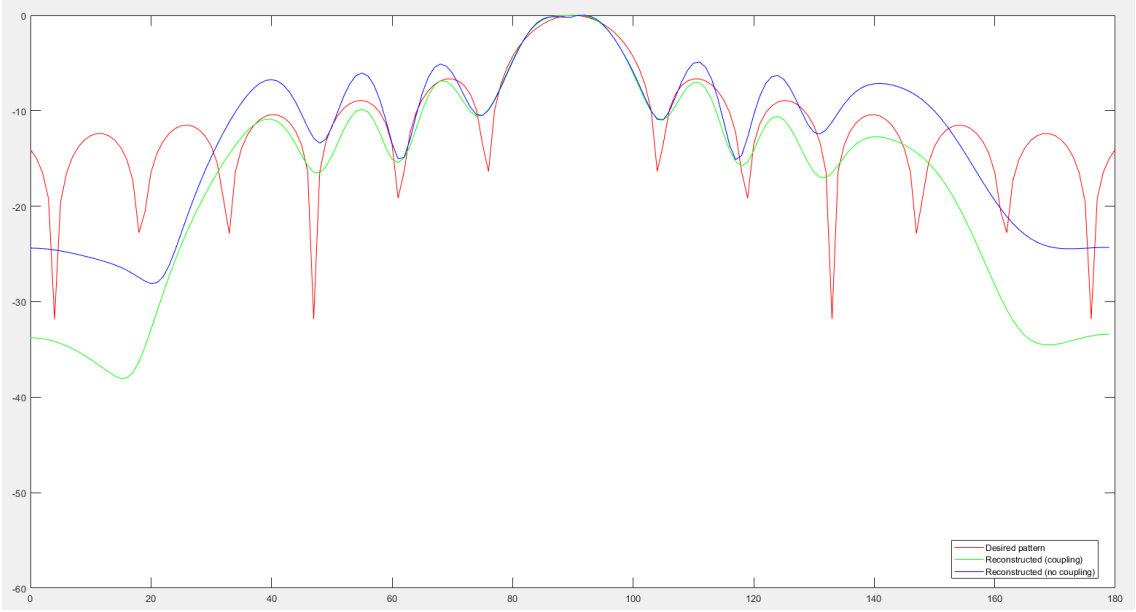

Figure 35: Fourier Transform solution for 21 elements on the x-z plane with a stretch factor of 4. Horizontal axis measures elevation angle in degrees.

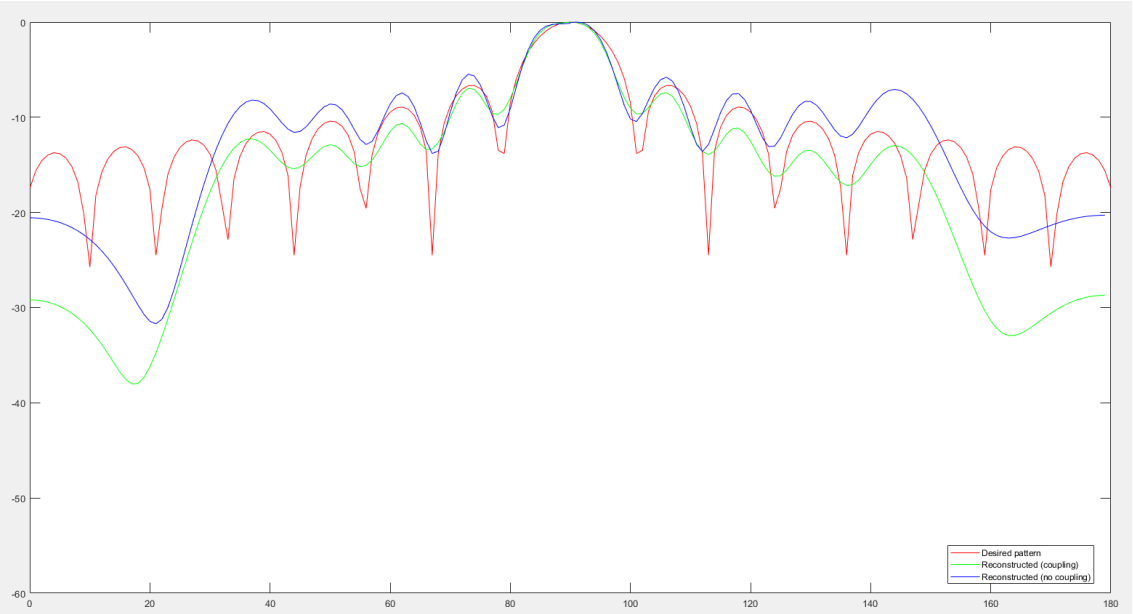

Figure 36: Fourier Transform solution for 21 elements on the x-z plane with a stretch factor of 5. Horizontal axis measures elevation angle in degrees.

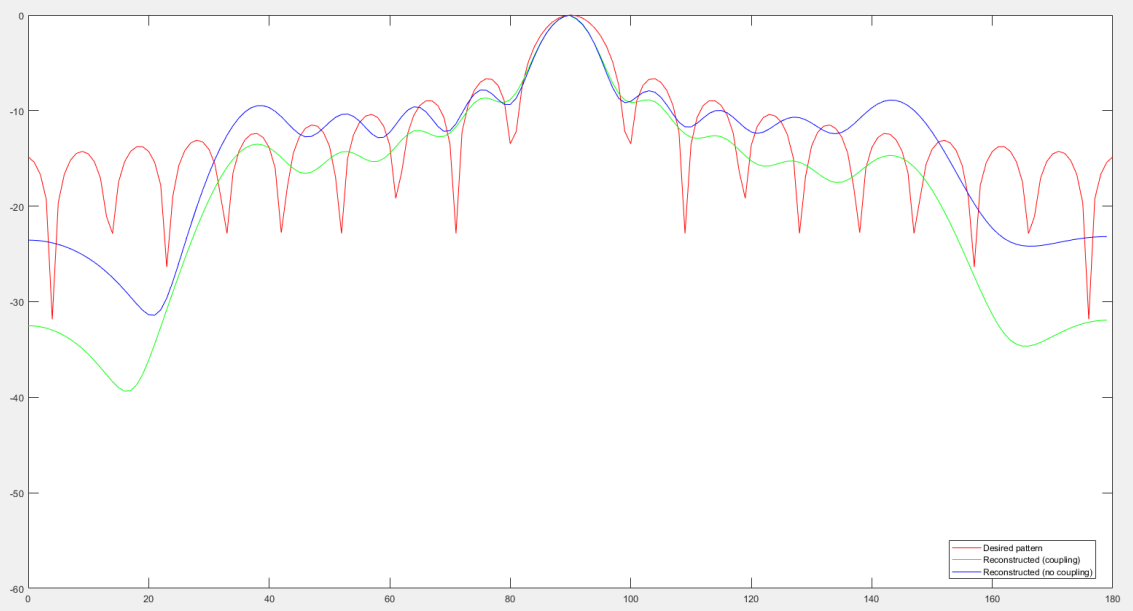

Figure 37: Fourier Transform solution for 21 elements on the x-z plane with a stretch factor of 6. Horizontal axis measures elevation angle in degrees.

By examining the main lobe in Figures 34, 35, 36, 37, a similar behavior is observed as expected, increasing the stretch factor, decreases the beamwidth. In Figure 34, with a

stretch factor of 3, the beamwidth is 22 degrees, while in figure 37, with a stretch factor of 6, the beamwidth is 14 degrees.

5 Conclusions and Future Work

5G networks are going to bring a radical change in the communications sector with all the new technologies enveloping them, as well as their enhanced features compared to previous communication architectures. Millimeter waves, antenna arrays, beamforming training algorithms, etc., contribute to the success of 5G, but also bring a number of challenges to overcome making them an active research field.

In this thesis, two pattern synthesis algorithms were developed in Matlab based on the linear system and the Fourier transform methods. As illustrated in the Results chapter, the Fourier transform algorithm showed more consistent and accurate results on both x-y and x-z planes, while the linear system algorithm showed no changes when more elements were added on the x-y plane and produced completely unsatisfactory results on the x-z plane.

Possible future work can be done on planar arrays, to compare them with the linear arrays using the same pattern synthesis algorithms. Lastly, the Wooodward-Lawson method can be examined and developed, in order to inspect if the results are more accurate than the linear system method and the Fourier transform method.

Bibliography

[1] J. Hampton, Introduction to MIMO Communications, Cambridge University Press, 2013.

[2] T. Marzetta, E. Larsson, H. Yang and H. Ngo, Fundamentals of Massive MIMO, Cambridge University Press, 2016.

[3] M. Agiwal, A. Roy and N. S. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey," *IEEE Commun. Surv. & Tuts.*, vol. 18, no. 3, pp. 1617-1655, Third Quarter 2016.

[4] F. Bocardi, R. W. Heath, A. Lozano, T. L. Marzetta and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74-80, February 2014.

[5] M. Arslan, K. Sundaresan and S. Rangarajan, "Software-defined networking in cellular radio access networks: Potential and challenges," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 150-156, January 2015.

[6] A. Osseiran et al., "Scenarios for 5G mobile and wireless communications: The vision of the METIS project," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 26-35, December 2014.

[7] E. Larsson, O. Edfors, F. Tufvesson and T. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 186-195, February 2014.

[8] R. Taori and A. Sridharan, "In-band, point to multi-point, mm-wave backhaul for 5G networks," *in Proc. IEEE Int. Conf. Commun. Workshops*, pp. 96-101, 2014.

[9] T. S. Rappaport et al., "Millimeter wave mobile communications for 5G cellular: It will work!," *IEEE Access*, vol. 1, pp. 335-345, May 2013.

[10] Z. Pi and F. Khan, "System design and network architecture for a millimeter-wave mobile broadband (MMB) system," *in Proc. IEEE Sarnoff Symp.*, pp. 1-6, 2011.

[11] S. Collonge, G. Zaharia and G. E. Zein, "Influence of the human activity on wide-band characteristics of the 60 GHz indoor radio channel," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2396-2406, November 2004.

[12] T. S. Rappaport, E. Ben-Dor, J. N. Murdock and Y. Qiao, "38 GHz and 60 GHz angle-dependent propagation for cellular & peer-to-peer wireless communications," *in Proc. IEEE Int. Conf. Commun.*, pp. 4568-4573, 2012.

[13] A. Ghosh et al., "Millimeter-Wave enhanced local area systems: A high-data-rate approach for future wireless networks,"," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1152-1163, June 2014.

[14] Q. C. Li, G. Wu and T. S. Rappaport, "Channel model for millimeter-wave communications based on geometry statistics," *in Proc. IEEE Globecom Workshops*, pp. 427-432, 2014.

[15] Q. Wu, J. Hirokawa, J. Yin, C. Yu, H. Wang and W. Hong, "Millimeter-wave multibeam endfire dual-circularly polarized antenna array for 5G wireless applications," *IEEE Trans. Antennas Propag.*, vol. 66, no. 9, pp. 4930-4935, September 2018.

[16] V. Kallnichev, "Analysis of beam-steering and directive characteristics of adaptive antenna arrays for mobile communications," *IEEE Antennas Propag. Mag.*, vol. 43, no. 3, pp. 145-152, June 2001.

[17] Rohde & Schwarz, "Millimeter-Wave Beamforming: Antenna Array Design Choices & Characterization," 2016.

[18] C. A. Balanis, Antenna Theory Analysis and Design, 4th ed., Wiley, 2015.

[19] S. Rajagopal, S. Abu-Surra, Z. Pi and F. Khan, "Antenna array design for multi-gbps mmwave mobile broadband communication," *in Proc. Global Telecommun. Conf. (Globecom)*, pp. 1-6, 2011.

[20] W. Roh et al., "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 106-113, February 2014.

[21] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao and J. I. Tamir, "Broadband millimeter wave propagation measurements and models using adaptive

beam antennas for outdoor urban cellular communications," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 1850-1859, April 2013.

[22] Y. M. Tsang, A. S. Y. Poon and S. Addepalli, "Coding the beams: Improving beamforming training in mmwave communication system," *in Proc. IEEE Global Telecommun. Conf.*, pp. 1-6, 2011.

[23] P. Xia, S. K. Yong, J. Oh and C. Ngo, "Multi-stage iterative antenna training for millimeter wave communications," *in Proc. IEEE Global Telecommun. Conf. (Globecom)*, pp. 1-6, 2008.

[24] Y. M. Tsang and A. S. Y. Poon, "Successive AoA estimation: Revealing the second path for 60 GHz communication system," *in Proc. IEEE Commun. Annu. Allerton Conf. Control Comput.*, pp. 508-515, 2011.

[25] Y. M. Tsang and A. S. Y. Poon, "Detecting human blockage and device movement in mmWave communication system," *in Proc. Global Telecommun. Conf. (Globecom)*, pp. 1-6, 2011.

[26] P. Xia, S. K. Yong, J. Oh and C. Ngo, "A practical SDMA protocol for 60 GHz millimeter wave communications," *in Proc. 42nd Asilomar Conf. Signals Syst. Comput.*, pp. 2019-2023, 2008.

[27] Mathworks, [Online]. Available: https://www.mathworks.com/help/antenna/ug/method-of-moments-metal-and-dielectric.html.

Appendix

Matlab code for the linear system algorithm on the x-y plane

```
close all;
clear variables;

c = physconst('lightspeed');
fc = 60e9;
lambda = c/fc;

dphi = 1;        % take a sample every dphi degrees
stretch = 3;     % increasing the value will decrease beam-
width
NumElements = 21;
ElementSpacing = 1.2*lambda/2;

el = design(patchMicrostripCircular,fc);
el.GroundPlaneWidth = inf;

la = lin-
earArray('Element',el,'NumElements',NumElements,'ElementSpa
cing', ...
     ElementSpacing);

figure; show(la);

azi_grid = 0:1:360;
elv_grid = 0;

[Ei,azii,elvi] = pat-
tern(el,fc,azi_grid,elv_grid,'Type','efield');

the_deg = 0:90;
AF = abs(sinc(stretch*(pi/2-the_deg*pi/180)));

figure; plot(the_deg,AF); title('Desired pattern (par-
tial)');

AF_complete = [AF fliplr(AF(2:end)) AF(2:end)
fliplr(AF(2:end))];

AF_reduced = AF_complete(1:dphi:end);
```

```
figure; plot(azi_grid,AF_complete); grid;
title('Desired pattern (complete)');

dphi_table = 0:dphi:360;

k = 2*pi/lambda;
theta = 90;
phi = 0;
v = 90 - theta;

for R = 1:length(dphi_table)
    for C = 1:NumElements
        B(R,C) = exp(1j*(C-
1)*k*ElementSpacing*cosd(v)*cosd(dphi_table(R)));
    end
end

X = B\AF_reduced';

figure;
subplot(2,1,1); stem(abs(X));
subplot(2,1,2); stem(angle(X)*180/pi);

Residues = AF_reduced'-B*X;

figure; title('Residual errors');
subplot(2,1,1); plot(abs(Residues));
subplot(2,1,2); plot(angle(Residues)*180/pi);

la.AmplitudeTaper = abs(X);
la.PhaseShift = angle(X)*180/pi;    % remember to make
PhaseShift in degrees

AFR_dB = arrayFactor(la,fc,azi_grid,elv_grid);  % this is
in dB, need to convert
AFR = 10.^(AFR_dB/10);

Etotalr = AFR.*Ei;

[DD,azi,elv] = pattern(la,fc,azi_grid,elv_grid);
Max_D = max(max(DD));
Max_AF_reduced = max(max(AF_reduced));
Max_E = max(max(Etotalr));

figure; polarpattern(azi,DD);

figure;
plot(azi(1:dphi:length(AF_complete)),10*log10(AF_reduced)-
10*log10(Max_AF_reduced),'r');
hold on; plot(azi,DD-Max_D); hold on;
```

```
plot(azi,10*log10(Etotalr)-10*log10(Max_E),'g');

legend('Desired pattern','Reconstructed (cou-
pling)','Reconstructed (no coupling)');
```

Matlab code for the linear system algorithm on the x-z plane

```
close all;
clearvars;

c = physconst('lightspeed');
fc = 60e9;
lambda = c/fc;

stretch = 3;       % increasing the value will decrease
beamwidth
dtheta = 1;        % take a sample every dtheta degrees
NumElements = 21;
ElementSpacing = 1.2*lambda/2;

el = design(patchMicrostripCircular,fc);
el.GroundPlaneWidth = inf;

la = lin-
earArray('Element',el,'NumElements',NumElements,'ElementSpa
cing', ...
     ElementSpacing);

figure; show(la);

azi_grid = [0 180];
elv_grid = 0:90;

[Ei,azii,elvi] = pat-
tern(el,fc,azi_grid,elv_grid,'Type','efield');

the_deg = 0:90;
AF = abs(sinc(stretch*(pi/2-the_deg*pi/180)));

figure; plot(the_deg,AF); title('Desired pattern (par-
tial)');

AF_complete = [AF fliplr(AF(2:end))];
AF_reduced = AF_complete(1:dtheta:end);

figure; plot(0:180,AF_complete); grid;
title('Desired pattern (complete)');
```

```
dtheta_table = 0:dtheta:180;

k = 2*pi/lambda;
v = [0:90 89:-1:0];    % elevation angle counting from
equator

phi = [zeros(1,91) repmat(180,1,90)];
v2 = v(1:dtheta:end);   % sampled elevation angle from
equator by dtheta
phi2 = phi(1:dtheta:end);

for R = 1:length(v2)
    for C = 1:NumElements
        B(R,C) = exp(1j*(C-
1)*k*ElementSpacing*cosd(v2(R))*cosd(phi2(R)));
    end
end

X = B\AF_reduced';

figure;
subplot(2,1,1); stem(abs(X));
subplot(2,1,2); stem(angle(X)*180/pi);

Residues = AF_reduced'-B*X;

figure; title('Residual errors');
subplot(2,1,1); plot(abs(Residues));
subplot(2,1,2); plot(angle(Residues)*180/pi);

figure; plo-
tyy(dtheta_table,AF_reduced,dtheta_table,abs(Residues));

la.AmplitudeTaper = abs(X);
la.PhaseShift = unwrap(angle(X)*180/pi);    % remember to
make PhaseShift in degrees

AFR_dB = arrayFactor(la,fc,azi_grid,elv_grid);  % this is
in dB, need to convert
AFR = 10.^(AFR_dB/10);

Etotalr = AFR.*Ei;

[DD,azi,elv] = pattern(la,fc,azi_grid,elv_grid);

Max_D = max(DD(:));
Max_AF_reduced = max(AF_reduced(:));
Max_E = max(Etotalr(:));

new_elv = [elv_grid 90+elv_grid(1:end-1)];
```

```
figure; plot(dtheta_table,10*log10(AF_reduced)-
10*log10(Max_AF_reduced),'r');
hold on; plot(new_elv, [DD(:,1)-Max_D; flipud(DD(1:end-
1,2)-Max_D)], 'g')
plot(new_elv, [Etotalr(:,1)-Max_E; flipud(Etotalr(1:end-
1,2)-Max_E)], 'b')
legend('Desired pattern','Reconstructed (cou-
pling)','Reconstructed (no coupling)','Location','Best');
```

Matlab code for the Fourier transform algorithm on the x-y plane

```
close all;
clearvars;

c = physconst('lightspeed');
fc = 60e9;
lambda = c/fc;

stretch = 3;    % increasing the value will decrease beam-
width
dphi = 1;       % take a sample every dphi degrees
NumElements = 21;
ElementSpacing = 1.2*lambda/2;

el = design(patchMicrostripCircular,fc);
el.GroundPlaneWidth = inf;

la = lin-
earArray('Element',el,'NumElements',NumElements,'ElementSpa
cing', ...
     ElementSpacing);

figure; show(la);

% azi_grid, elv_grid contain the standard angles where the
directivity
% should be computed (with coupling)
azi_grid = 0:180;
elv_grid = 0;

% azi_grid_nc, elv_grid_nc should contain intervals to be
used for computing pattern without
% coupling (variable name _nc), where we need additional
angles to the ones of interest
azi_grid_nc = 0:180;
elv_grid_nc = 0:5:90;

% pattern of individual element
```

```
[Ei,azii,elvi] = pat-
tern(el,fc,azi_grid_nc,elv_grid_nc,'Type','efield');

phi_deg = 0:90;
Pat = abs(sinc(stretch*(pi/2-phi_deg*pi/180)));

% complete pattern constructed by replication
Pat_complete = [Pat fliplr(Pat(1:end-1))];

% sampled version of Pat_complete
Pat_reduced = Pat_complete(1:dphi:end);

figure;
subplot(2,1,1); plot(phi_deg,Pat); title('Desired pattern
(partial)');
subplot(2,1,2); plot(0:180,Pat_complete); grid; ti-
tle('Desired pattern (complete)');

% proper cut of individual pattern for x-y plane (i.e.
elvi=0)
E_cut = Ei(elvi==0,:);

% recall that Pat_reduced has units of power while |Etot|
must be E-field,
% so we have to take sqrt. Also, we NEED to make sure that
Pat_reduced is
% in linear scale and not dB
Etot = sqrt(Pat_reduced);

% this corresponds to |Etot|/|Eind|=|AF|, sampled at the
angles of interest
RHS = Etot./E_cut(1:dphi:end);
assert(isempty(find(imag(RHS)~=0, 1)));   % RHS should be
have no imag part

dphi_table = 0:dphi:180;
figure; plot(dphi_table,RHS); title('Desired |AF|'); xla-
bel('degrees in y-z plane');

beta = 0;

k = 2*pi/lambda;

% xia is the angle formed on the x-z plane between the pos-
itive x axis and
% the direction of interest
xia = 0:dphi:180;

% don't use a named variable psi because Matlab has a func-
tion named psi!
psi1 = 2*pi/lambda*ElementSpacing*cosd(xia)+beta;
```

```
dpsi = psi1(2)-psi1(1);
cur = zeros(1,NumElements);

% truncate |AF| into interval from -pi to pi
RHS_trunc = RHS(abs(psi1)<=pi);
psi_trunc = psi1(abs(psi1)<=pi);

figure; plot(psi1,RHS); hold on;
plot(psi_trunc,RHS_trunc,'r');
title('|AF(psi)|'); xlabel('radians');
legend('Regular','Truncated');

M = (NumElements-1)/2;
arr_index = -M:M;

for m = 1:NumElements
   cur(m) = sum(exp(-
1j*arr_index(m)*psi_trunc).*RHS_trunc)*dpsi/(2*pi);
end

figure;
subplot(2,1,1); stem(abs(cur));
subplot(2,1,2); stem(angle(cur)*180/pi);

la.AmplitudeTaper = abs(cur);
la.PhaseShift = angle(cur)*180/pi;    % remember to make
PhaseShift in degrees

[AFR_dB,azi_AF,elv_AF] = arrayFac-
tor(la,fc,azi_grid_nc,elv_grid_nc);  % this is in dB, need
to convert to linear
AFR = 10.^(AFR_dB/10);

Et = AFR.*Ei;
Et = Et.^2;  % need |E|^2 for radiation intensity U (check
Balanis)

dthD = (elv_grid_nc(end)-
elv_grid_nc(1))/(length(elv_grid_nc)-1)*pi/180;  % convert
to radians
dphiD = (azi_grid_nc(end)-
azi_grid_nc(1))/(length(azi_grid_nc)-1)*pi/180; % convert
to radians
Prad = 0;

for ind = 1:length(azi_grid)
   Prad = Prad+
dot(Et(:,ind),cosd(elv_grid_nc))*dthD*dphiD;
end
```

```
Et_dB = 10*log10(Et*4*pi/Prad);

[DD,azi,elv] = pattern(la,fc,azi_grid,elv_grid);
Max_D = max(DD(elv==0,:));
Max_Pat_reduced = max(Pat_reduced(:));
Max_Et_dB = max(Et_dB(elv==0,:));
Max_RHS = max(RHS(:));
Max_AFR = max(AFR_dB(elv_AF==0,:));

figure; plot(dphi_table,10*log10(Pat_reduced)-
10*log10(Max_Pat_reduced),'r');
hold on; plot(azi,DD(elv==0,:)-Max_D);
plot(azi,Et_dB(elv==0,:)-Max_Et_dB,'g');
legend('Desired pattern','Reconstructed (cou-
pling)','Reconstructed (no coupling)','Location','Best');

figure; plot(dphi_table,10*log10(RHS/Max_RHS)); hold on;
plot(azi_AF,AFR_dB(elv_AF==0,:)-Max_AFR); title('Array fac-
tors');
legend('Desired','Reconstructed');
```

Matlab code for the Fourier transform algorithm on the x-z plane

```
close all;
clearvars;

c = physconst('lightspeed');
fc = 60e9;
lambda = c/fc;

stretch = 3;   % increasing the value will decrease beam-
width
dtheta = 1;    % take a sample every dtheta degrees
NumElements = 21;
ElementSpacing = 1.2*lambda/2;

assert( floor(dtheta)==dtheta );

el = design(patchMicrostripCircular,fc);
el.GroundPlaneWidth = inf;
la = lin-
earArray('Element',el,'NumElements',NumElements,'ElementSpa
cing', ...
     ElementSpacing);

figure; show(la);

azi_grid = [0 180];
elv_grid = 0:90;
```

```
% pattern of individual element
[Ei,azii,elvi] = pat-
tern(el,fc,azi_grid,elv_grid,'Type','efield');

the_deg = 0:90;
Pat = abs(sinc(stretch*(pi/2-the_deg*pi/180)));

% complete pattern constructed by replication
Pat_complete = [Pat fliplr(Pat(1:end-1))];

% sampled version of Pat_complete
Pat_reduced = Pat_complete(1:dtheta:end);

figure;
subplot(2,1,1); plot(the_deg,Pat); title('Desired pattern
(partial)');
subplot(2,1,2); plot(0:180,Pat_complete); grid; ti-
tle('Desired pattern (complete)');

% proper cut of individual pattern for x-z plane
E_cut = [Ei(:,azii==0); flipud(Ei(1:end-1,azii==180))];

% recall that Pat_reduced has units of power while |Etot|
must be E-field,
% so we have to take sqrt. Also, we NEED to make sure that
Pat_reduced is
% in linear scale and not dB
Etot = sqrt(Pat_reduced);

% this corresponds to |Etot|/|Eind|=|AF|, sampled at the
angles of interest
RHS = Etot'./E_cut(1:dtheta:end);
assert(isempty(find(imag(RHS)~=0, 1)));   % RHS should be
have no imag part

dtheta_table = 0:dtheta:180;
figure; plot(dtheta_table,RHS); title('Desired |AF|'); xla-
bel('degrees in x-z plane');

beta = 0;

v = [0:90 89:-1:0];    % elevation angle counting from
equator
phi = [zeros(1,91) repmat(180,1,90)];
v2 = v(1:dtheta:end);   % sampled elevation angle from
equator by dtheta
phi2 = phi(1:dtheta:end);

% xia is the angle formed on the x-z plane between the pos-
itive x axis and
```

```
% the direction of interest
xia = 0:dtheta:180;

% don't use a named variable psi because Matlab has a func-
tion named psi!
psi1 = 2*pi/lambda*ElementSpacing*cosd(xia)+beta;
psi2 =
2*pi/lambda*ElementSpacing*cosd(v2).*cosd(phi2)+beta;

assert(isequal(psi1,psi2));

figure; plot(psi1,RHS); title('|AF(psi)|'); xla-
bel('radians');

dpsi = psi1(2)-psi1(1);
cur = zeros(1,NumElements);

% truncate |AF| into interval from -pi to pi
RHS_trunc = RHS(abs(psi1)<=pi);
psi_trunc = psi1(abs(psi1)<=pi);

M = (NumElements-1)/2;
arr_index = -M:M;

for m = 1:NumElements
   cur(m) = sum(exp(-
1j*arr_index(m)*psi_trunc)*RHS_trunc)*dpsi/(2*pi);
end

k = 2*pi/lambda;

figure; set(gcf,'name','Excitation current');
subplot(2,1,1); stem(abs(cur));
subplot(2,1,2); stem(angle(cur)*180/pi);

la.AmplitudeTaper = abs(cur);
la.PhaseShift = angle(cur)*180/pi;

AFR_dB = arrayFactor(la,fc,azi_grid,elv_grid);  % this is
in dB, need to convert
AFR = 10.^(AFR_dB/10);

Etotalr = Ei.*AFR;
Etotalr = Etotalr.^2; % need |E|^2 for radiation intensity
U (check Balanis)

dth = (elv_grid(end)-elv_grid(1))/(length(elv_grid)-
1)*pi/180;  % convert to radians
dphi = (azi_grid(end)-azi_grid(1))/(length(azi_grid)-
1)*pi/180; % convert to radians
Prad = 0;
```

```
for ind = 1:length(azi_grid)
   Prad = Prad+
dot(Etotalr(:,ind),cosd(elv_grid))*dth*dphi;
end

Et = 10*log10(Etotalr*4*pi/Prad);

[DD,azi,elv] = pattern(la,fc,azi_grid,elv_grid);

Max_D = max(DD(:));
Max_Pat_reduced = max(Pat_reduced(:));
Max_E = max(Et(:));

new_elv = [elv_grid 90+elv_grid(1:end-1)];

figure; plot(dtheta_table,10*log10(Pat_reduced)-
10*log10(Max_Pat_reduced),'r');
hold on; plot(new_elv, [DD(:,1)-Max_D; flipud(DD(1:end-
1,2)-Max_D)], 'g')
plot(new_elv, [Et(:,1)-Max_E; flipud(Et(1:end-1,2)-Max_E)],
'b')
legend('Desired pattern','Reconstructed (cou-
pling)','Reconstructed (no coupling)','Location','Best');
set(gca,'ylim',[-60 0]);
```