THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Modeling and Analysis of Millimeter-Wave Frequency Scanned Antenna Array

ANDREW SMITH
SUMMER 2022

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Electrical Engineering
with honors in Electrical Engineering

Reviewed and approved* by the following:

Dr. Wooram Lee
Associate Professor
Thesis Supervisor

Dr. Julio Urbina
Associate Professor
Honors Adviser

ABSTRACT

The implementation of a frequency scanned antenna array architecture to millimeter wave systems allows for a variety of benefits in RF system design. This thesis looks to implement a frequency scanned antenna array design with an input frequency scanning from 25-35 GHz and LO frequency of 120 GHz. Using a simple divider circuit, we are able to create a simplistic system architecture with four antennas. Simulations for this frequency scanned array are done via MATLAB SIMULINK with the RF BLOCKSET model in order to get an accurate and holistic view of its operation. The comprehensive simulations within MATLAB will allow future users to experiment and benchmark their designs of frequency scanned antenna arrays. The simulations will also allow for the development of a printed circuit board design of our specific frequency scanned antenna array. Overall, the applications of this frequency scanned array antenna can be utilized for designing high-quality radar sensing antennas.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to first thank Dr. Wooram Lee for supervising me on this thesis. Without his assistance, constructive criticism, and honesty this would not have been possible. I would also like to thank Dr. Julio Urbina for giving me the encouragement to continue and fostering my interest in Electromagnetics and RF.

Chapter 1

Introduction

With each successive generation radar and telecommunication gets pushed to another level of data usage. More people are using the internet and wireless services on a scale inconceivable a 30 years ago. Wireless communications and sensors have become integrated into almost every device. As a result of this increased demand is the need for faster transmission and a greater bandwidth. 5G millimeter wave (mmWave) technology hopes to remedy this situation we find ourselves in. The mmWave band has a greater bandwidth for data transmission and higher speeds. The only issue is the range of mmWave and in the realm of telecommunication link discovery.

To deal with the short comings of mmWave, engineers in the realm of phased array antenna design have developed systems and architectures for faster link discovery and transmission time. Many of this technology is currently being adapted already with mmWave. For example, phased antenna array sensors with mmWave bands have been introduced in burgeoning realm of self-driving and autonomous vehicles. Not to mention the new link discovery method of single shot, in which there are multiple beams that are sent from the device to discover the connection [1].

Phased Array Antennas have been the most popular implementation of mmWave arrays and sensors. However, they come with considerable issues. For one the phased array antennas are harder and more costly to manufacture as they have a greater level of complexity to them with phase shifters on the end of every antenna in a phased array. This heavier architecture

of the system in turn makes it more difficult to specifically beam steer and ends up taking longer. Also phased array antennas are much less feasible for the single shot method of link discovery needed to improve 5G. When it comes to transmitting multiple signals the phased array antenna has a great difficulty scaling the number of signals sent.

To remedy the short comings of a phased array antenna, an alternative frequency scanned array design is put in place. Frequency scanned arrays instead of sweeping the phase of each antenna element sweeps the frequency of the systems which through the transmission line structure creates a phase shift [2]. This instantaneous change with the phase shift relative to the frequency allows a faster steering of the beam pattern. Along with that, the simple transmission line architecture minimizes the cost and complexity of the system overall [3]. Lastly, single shot method is more feasible and scaling up and sending beams is much easier.

As stated before, phased array antennas have been the technology of choice for a variety of mmWave and high frequency technologies. Many of these phased antenna arrays have various designs but they rely on one key element: the phase shifter. The phase shifters were added to the ends of the antennas to induce a phase shift. When these phase shifts were summed up, they were able to produce a beam that was able to be electronically steered. In Figure 1 below there is an example of a phased array system where there is large common beam pattern made from the steering of the antenna.

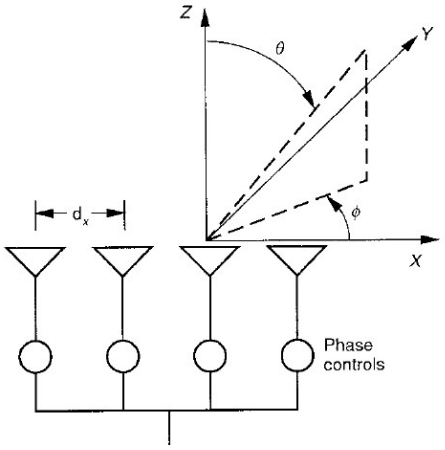


Figure 1: Phased Array Design[1]

During the early development of antenna arrays, frequency scanned arrays were also being looked into as a model for beam steering. In back in the 1960's Ishimar and H.S. Tuan developed the first frequency scanned array antenna theory[2]. This revolutionary paper laid the mathematical and physical foundations to develop frequency scanned antenna arrays. They followed a specific model known as the serpentine line in which delays inherent in the coaxial line allowed for a shift in the angle of the propagating wave from the antenna. Each antenna was placed at these shifts and was able to develop into a frequency scanned array with the two calculated spacings of a and a' seen in Figure 2.

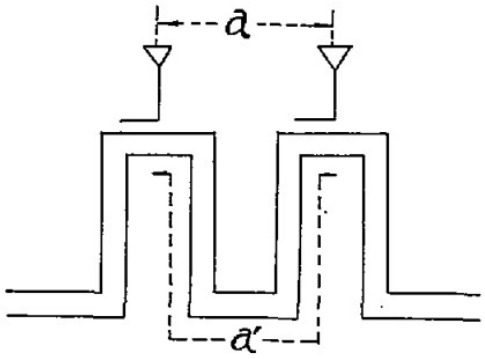


Figure 2: Early Serpentine Line Design of Frequency Scanned Arrays[2]

A frequency scanned array operating at frequencies from 145-155 GHz is designed using MATLAB Simulink. This design should give a better instance on how frequency scanned arrays work and a lot of useful parameters to benchmark and test. This is to also encourage other designers when testing out frequency scanned array antenna architectures for various 5G and radar projects.

Chapter 2

Phased Array Theory and Modeling

2.1 Introduction

Phased array theory is the basis for the Frequency Scanned Array design. To design an efficient frequency scanned array system, the fundamentals of the beampattern and the Dolph-Chebyshev weights for sidelobe suppression must be determined. Each of these principles are outlined in the *Phased Array Antenna Handbook* for each of these equations [12].

2.2 Derivation of Beampattern

To extract the phase shift between elements the following equation is derived as each angle is subtracted by the adjacent angle.

$$\Delta\theta = \theta_{n+1} - \theta_n \qquad (2.1)$$

Following the shift in angles, it is now important to set up the given values that describe the shift in the antenna array within the system in Figure 1. The angular wavenumber (k) along with the spacing (d) describe the shifting of the individual elements with respect to the variable angle input (θ). This is added to the phase shift to get a complete view of how each antenna shifts.

$$\varphi(\theta) = kdsin\theta + \Delta\theta;\ \ k = 2\pi/\lambda \qquad (2.2)$$

Using the $\varphi(\theta)$ equation, it is time to extract the Array Factor equation (AF). The AF gives a complete signal on the magnitude of the beam and what specific angle it is most intense.

Each antenna is modelled by $e^{jm\varphi}$ as a propagating RF signal. Following this, the each of the signals are summated together to get the following equation.

$$AF(\varphi) = \sum_{m=0}^{N-1} e^{jm\varphi} = 1 + e^{j\varphi} + e^{j2\varphi} + \cdots + e^{j(N-1)} \qquad (2.3)$$

The AF will allow us an insight into the beampattern of the entire system. These beampattern plots are done in various graphs such as the polar plots and a cartesian one. Important parameters can be determined from such plots such as the side lobe suppression and the beamwidth of the main lobe. All of these parameters are in turn used to develop a cohesive system.

2.3 Dolph-Chebyshev Window

To strengthen the frequency scanned array antenna's accuracy, a beam scanning windowing function must be designed. Windowing algorithms for frequency scanned arrays reduce the side lobes that follow many of the beampatterns in the array factor output. Through the following Dolph-Chebyshev Window algorithm we can determine the side lobe suppression (SSL) weights at specified decibel values and element values.

The Dolph-Chebyshev algorithm is a filtering technique utilized in array antenna design to create a window to attenuate the main beam signal and suppress unnecessary signals. For this application, the side lobes of our antenna array are undesirable and must be filtered out while in turn not tapering the main lobe. A unique property of the Dolph-Chebyshev algorithm is its ability to mostly flatten the side lobes at a relatively uniform decibel level.[5]

To start the Dolph-Chebyshev algorithm, there must be a conversion from the designated SLL in decibels to the voltage ratio R_0.

$$R_0 = 10^{SLLdB/20} \qquad (2.4)$$

Following this conversion, there must be a subsequent decision to set the number of elements N to m which will be used in the Chebyshev polynomial later.

$$m = N - 1 \qquad (2.5)$$

The Chebyshev polynomial $T_m(z)$ consists of the input value z which is specifically equal to cos(θ). As the input z is cos(θ) the following equation input is bounded from a maximum and minimum value of 1 and -1 respectively. Within the polynomial it is bounded by the only available inputs of $cosh^{-1}$. So z must be greater than 1 or less than 1.

$$T_m(z) = \cosh[m * cosh^{-1}(z)] \qquad z < -1, z > 1 \qquad (2.6)$$

The Chebyshev polynomial will be rearranged to obtain the z_0 which will be instrumental in determining the zeros of the system at specific phases (φ_i). The $z_0$value is set through equating $T_m(z_0) = R_0$ and the resulting equation is found below.

$$z_0 = \cosh\left[\frac{1}{m} cos^{-1}\big(R_0(VR)\big)\right] \qquad (2.7)$$

Following the discovery of z_0 equation, Chebyshev zeroes(z_i) of the polynomial must be determined. Since the Chebyshev polynomial is based on the cosine function, z_i is a product of the number of the following cosine formula.

$$z_i = \cos\frac{2i-1}{2m}; \;\; i = 1,2,3, \dots m \qquad (2.8)$$

Now that the parameters of z_i and z_0 have been determined, the phase φ_i where the zeroes occur can be derived. Again, $z = \cos(\theta)$ and the relation between φ_i and θ is $\theta = \frac{\varphi_i}{2}$. The z value can also be described by $z = \frac{z_i}{z_0}$. After substituting all these values into one $z = \cos(\theta)$ equation, the final step to derive φ_i is to find the inverse. As a result, the following equation is derived for φ_i:

$$\varphi_i = 2cos^{-1}\left(\frac{z_i}{z_0}\right);\ i = 1,2,3,\dots m \quad (2.9)$$

Each specific phase was put into a modified polynomial array factor equation following the Chebyshev polynomial equation. In the polynomial array factor equation, each element a_N is a specific coefficient value that attenuates or depreciates the specific value. These specific values are in a binomial expansion pattern, increasing until reaching the midpoint and then symmetrically decreasing.

$$AF = \prod_{i=1}^{m}(z - e^{j\varphi_i}) = 1 + a_1 z + a_2 z^2 + \cdots + a_{N-1} z^{N-1} \quad (2.10)$$

The output of each a_N value is from 1 to a higher weight greater than 1. In order to, normalize this there is an added normalization equation for each coefficient. Where the center coefficient is divided by each so the greatest coefficient is 1 and every other coefficient is less than 1.

$$a_{norm} = \frac{a_N}{a_{center}} \quad (2.11)$$

The normalized coefficients for a Dolph-Chebyshev window for a 6-element antenna array with -22dB of SLL should be the following: $a_1 = 0.4683$, $a_2 = 0.7559$, $a_3 = 1$, $a_4 = 1$, $a_5 = 0.7559$, and $a_6 = 0.4683$. To reiterate, from these specific coefficients multiplied by each of the antenna element's Array Factor values the specific SLL is possible and can be further implemented with RF amplifiers at different gains corresponding to the Dolph-Chebyshev coefficients.

There are various properties that should be kept in mind when modeling with the Dolph-Chebyshev Filter. First off, the beamwidth of the signal is largely disproportionate to the relative length or spacing of each antenna. In Figure 3 there is a plot demonstrating this property through a testing of the approximate value, Elliot equation, and the exact values.

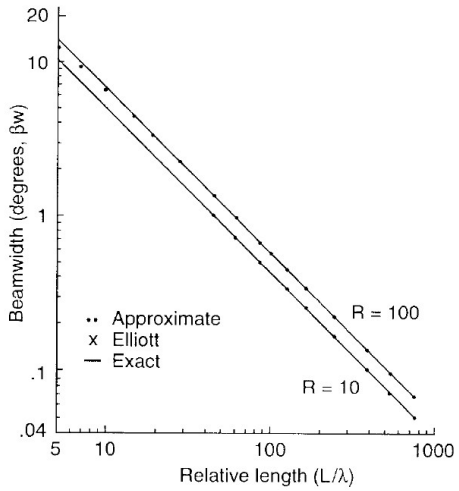


Figure 3: Beamwidth vs Relative Length of Antenna Space[12]

Another important property is how the Array Directivity is not always linearly increasing with the length of length of spacing. Below in Figure 4 there is a demonstration of how at a lower SSL there is limit.

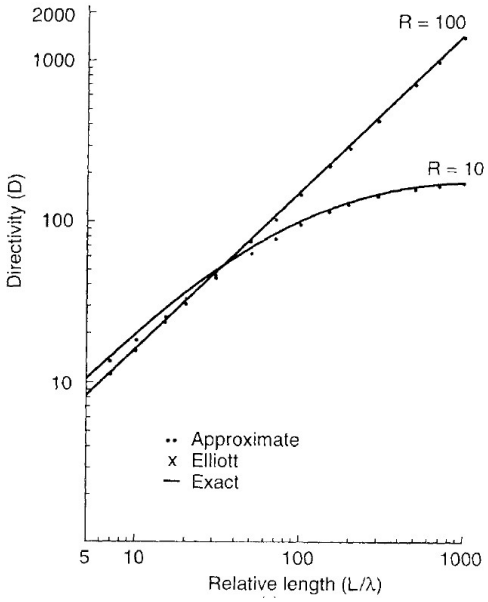


Figure 4 : Dolph Chebyshev Pattern Array Directivity vs Length [12]

As a result of this analysis, it would be good to model this carefully, and especially work towards having a greater SSL value if the spacing is much greater

Analysis of the Beamwidth is utilized through developing the phase array calculation equation below:

$$\theta_h = \cos^{-1}(\cos(\frac{\pi}{2}) - (0.433\left(\frac{\lambda}{L+d}\right))) - \cos^{-1}(\cos\left(\frac{\pi}{2}\right) + (0.433\left(\frac{\lambda}{L+d}\right))) \quad (2.12)$$

However, another factor to put in is the broadening factor that is a result of the Dolph-Chebyshev tapering suppression.

$$f = 1 + 0.636(2R_o^2)\cosh(((\cosh^{-1}(R_0)\text{^}2 - \pi^2)\text{^}0.5))\text{^}2 \quad (2.13)$$

Thus this all culminates in the following beamwidth formula.

$$Beamwidth = \theta_h f \quad (2.14)$$

.

Chapter 3

Proposed Frequency Scanned Array and Theory

3.1 Frequency Scanned Array Model

Frequency scanned arrays have individual shifts in the line that allow for it to induce a specific phase shift at the output. This means there is a power divider circuit that is generally constructed around this system. Below in Figure 5 is the specific frequency scanned array model that the system I will be designing will be based off. This specific diagram was designed with the help of the HSIC program at Penn State. It is an original frequency scanned array device scanning from the 145GHz to 155GHz.

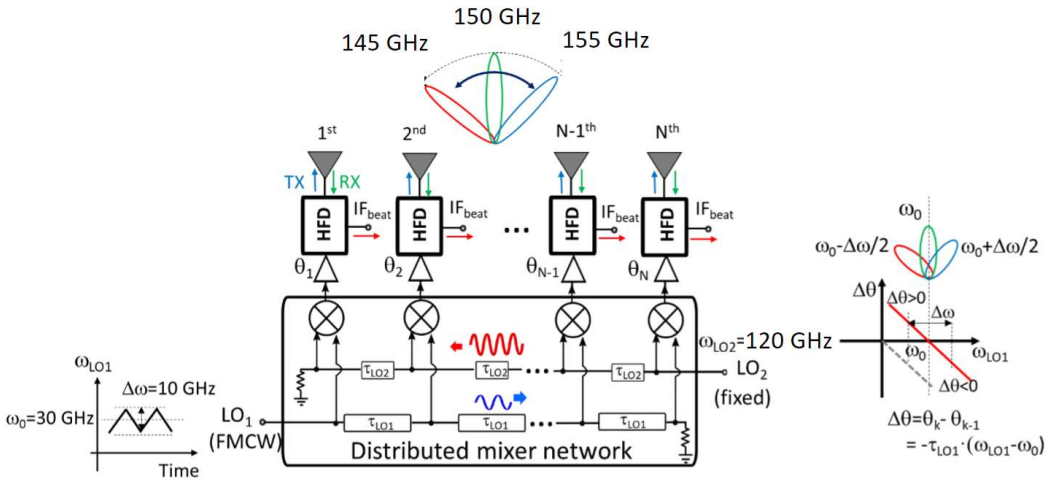


Figure 5 : Frequency Scanned Array Model

In the Figure 5 we can see the tau transmission line divider network for the frequency scanned array [4]. Each one of them will induce a phase shift at a determined

frequency value [5]. Overall, this model will have a LO frequency input of 120GHz and an IF Frequency input from 25GHz to 35GHz. Each of these sources has a divider network to which subsequently feeds into the mixer and rest of the antenna. This design will model and be the basis of the rest of the model as we move to the equations that determine the values for the system.

3.2 Derivation of Steering Angle

To derive the steering angle, there must first be an understanding of phi of each antenna element k. This k is bounded from 1 to N-1 as that is the denotation of each antenna in the antenna array theory. There is a dela tau and angular frequency attached to each of the k-1 and N-k values that are subsequently subtracted by one another to give us the phi angle for each element.

$$\varphi_k = -(k-1)\tau_{LO1} * \omega_{LO1} - (N-k)\tau_{LO2} * \omega_{LO2}\ (1 < k < N+1) \qquad (3.1)$$

Moving on from this, there we can assume due to the matching of the circuits that delays and angular frequency are equal as we wish to achieve this matching in our circuit design which results in the following equation.

$$\tau_{LO1} * \omega_{LO1} = \tau_{LO2} * \omega_{LO2} \qquad (3.2)$$

Following this, we can obtain the phase shift of phi of subsequent antenna elements and using the previous equations we can substitute in the values and arrive at $\tau_{LO1} * \Delta\omega$.

$$\Delta\varphi = \varphi_k - \varphi_{k-1} = -\tau_{LO1} * \omega_{LO1} + \tau_{LO2} * \omega_{LO2} = \tau_{LO1}(\omega_0 - \omega_{LO1}) = \tau_{LO1} * \Delta\omega$$

(3.3)

We also know from Antenna array theory that each shift is also modeled by the $\sin(\theta) * d * \left(2 * \frac{\pi}{\lambda}\right)$ so we set the following equal to the negative of $\Delta\varphi$ since it is the reverse.

$$\sin(\theta) * d * \left(2 * \frac{\pi}{\lambda}\right) = -\tau_{LO} \ * \Delta\omega \quad (3.4)$$

From antenna array theory we also know that lambda can result be simplified to the following equation with the speed of light modeled for c .

$$\lambda = 2\pi * \frac{c}{\omega_0 + \omega_{LO}} \qquad (3.5)$$

Finally, we can substitute all of the values in and arrive at the given isolating the $\sin(\theta)$ and thus getting the steering angle by taking the inverse sin of sin to give us θ the steering angle.

$$\theta = \arcsin\left(-\frac{\tau_{LO1} * \Delta\omega}{\pi} \frac{\tau_{LO2} * \omega_{LO2}}{\tau_{LO} \ * \Delta\omega + (\tau_{LO} \ + \tau_{LO2}) * \omega_{LO}}\right) \qquad (3.6)$$

Knowing this we can simulate the and have a theoretical model to control the steering angle of our frequency scanned array.

Chapter 4

Modeling and Design of Frequency Scanned Array in MATLAB

4.1 Introduction

MATLAB Simulink is a block diagram environment with the MATLAB software used to simulate various designs without using code. Simulink is designed to be intuitive and more efficient to understand simulating complex systems. The Simulink design program has various plugins functions that can be used to model various pieces of complex RF system architecture within its system. For this specific project, the specific plugin that will be utilized will be the RF Blockset toolbox. The RF blockset toolbox in particular will take preeminence over this design as it provides a comprehensive library of RF components necessary to develop a frequency scanned array system . RF Blockset has various mixers, amplifiers, and Wilkinson dividers needed to test this specific circuit.

4.2 Overall Simulink Design

The overall simulation system model is divided into five main blocks as shown in Figure 6. First off, there is the source block which consists of both LO frequency input and the IF frequency input sources. The other four blocks are all antenna array systems which have different numbered antenna elements. The first antenna system has four elements, the second system consists of eight elements, the third system consists of sixteen elements, and the final

system has thirty-two elements. This specific section makes up a bulk of the simulation data and since each specific system cannot run at the same time they are commented in and out.

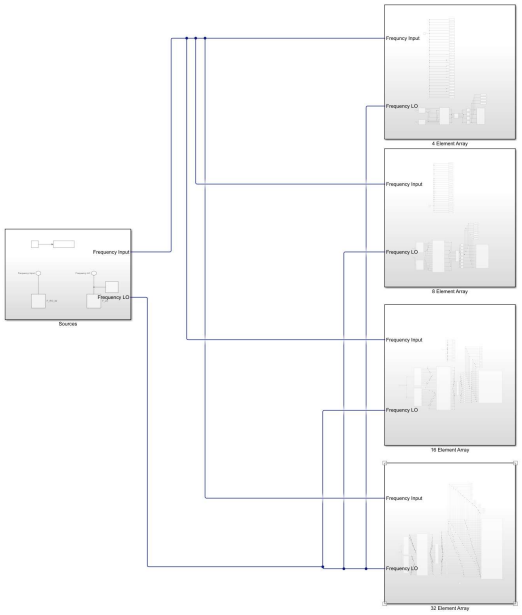


Figure 6 : Overall Simulink Architecture

As this system is based off of the one developed in the background in theory section it in turn abides by several global variables in order to create the beam steering system. Within table 1 there are the key global variables used for the frequency scanned array system.

Table 1 : Global Simulation Variables

Variable	Value
F_IF	25 GHz - 35 GHz
F_LO	120 GHz
Tau_IF	0.07 ns
Tau_LO	0.0175 ns

Each of these specific variables will govern the beam steering of this frequency scanned array. The Tau_IF and Tau_LO variables will also be implemented in many cases as multiples of its base value for the divider circuit.

4.3 RF Signal Source Block

For the power supply block there are several specific components that make up this system. Both power supplies for frequency are utilizing the sinusoidal source block which. Each specific sinusoidal source is given an amplitude of 40V and respective frequency values for LO and IF. The IF frequency in the variable F_IF is specifically modified after each specific simulation in between the values of 25GHz and 35GHz. In the power supply block below there is another specific instance called the RF configuration block which sets up key aspects of the simulation. In this specific block, each step size of data is 0.16*10^-7 microseconds creating a system bandwidth of 62500GHz. This sampling is more than enough data needed to specifically capture everything needed at high frequencies. Along with that, there is a specific temperature

setting for the configuration which is kept at a normal 290K. Lastly, as a part of this system there is a value outputting the input frequency used in the simulation to the analysis section.

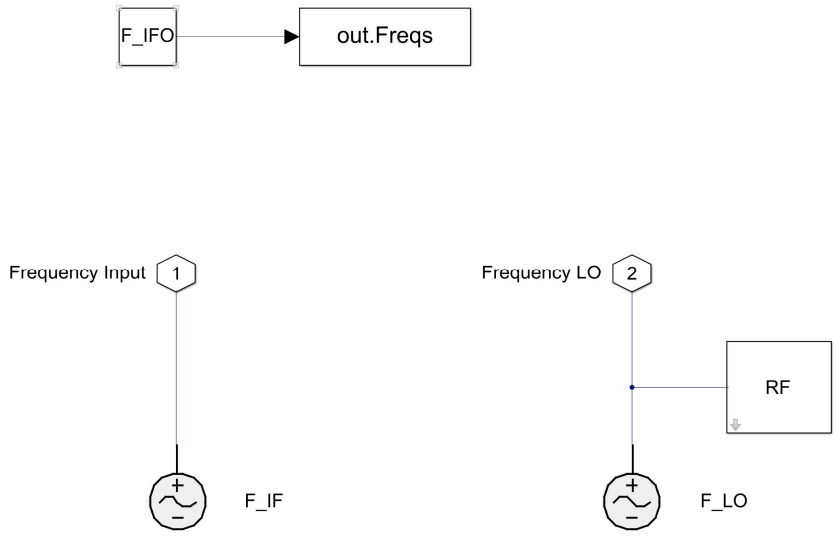
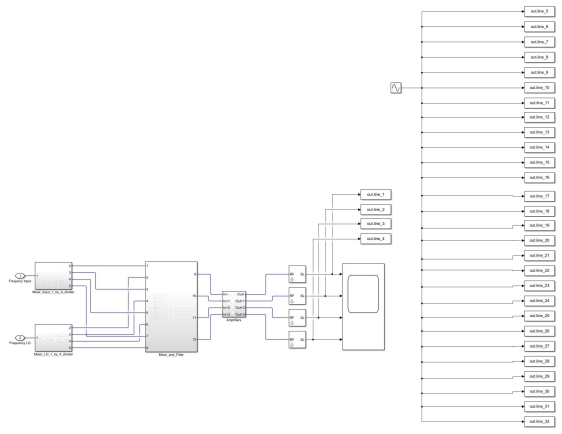


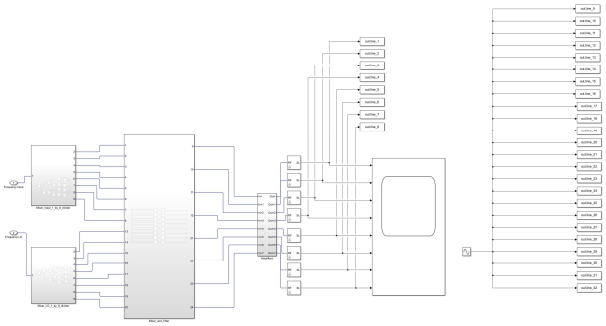
Figure 7: Power Supply Block

4.4 Element Arrays : Four Through Thirty-Two

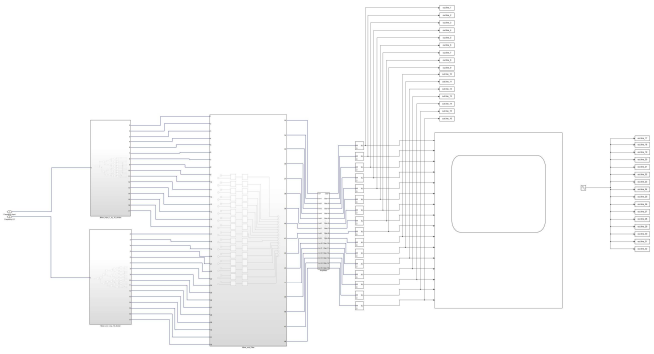
With all the antenna element arrays there are four specific blocks. Each of these specific blocks and their components are scaled up and down relative to the number of elements in their respective array antenna system. Each specific part of this system feeds from the divider circuit to the outport blocks. This starts with the IF Frequency Divider circuit and Lo Frequency Divider circuit which then feeds into the Mixer and Filter circuit. Subsequently, the Mixer and Filter circuit feeds into the Amplifier Block and out of the system. At the end of each antenna system there is an outport block which converts the RF Blockset circuit envelope data type to a Real passband data type which is available for further data analysis. In each specific outport there is a sampling rate that needs to be defined which for the model is kept at 1*10^-15 s.



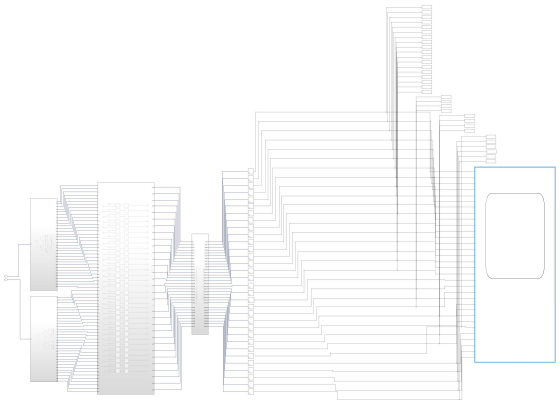
(a) Four Element Antenna Architecture



(b) Eight Element Antenna Architecture



(c) Sixteen Element Antenna Architecture



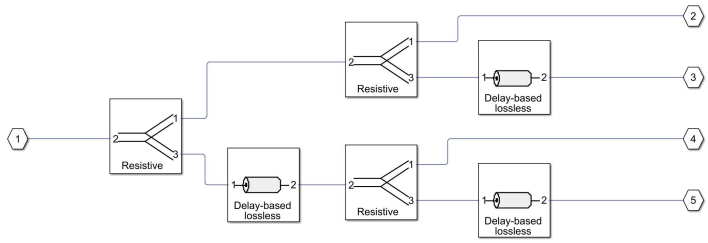
(d) Thirty-Two Element Antenna Architecture

Figure 8 : All Antenna Array Architecture Block Diagrams

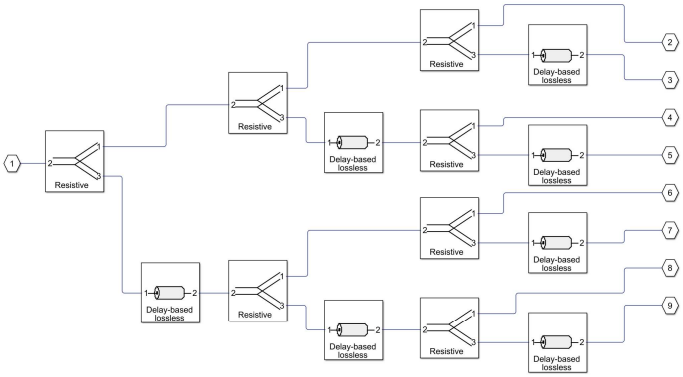
Mixer IF and LO Dividers

Both the IF and LO Frequency divider circuits are designed to create delays within the system that will produce the beam steering capabilities. Both blocks consist of the two basic components the Wilkinson Divider and delay based lossless transmission lines. Each Wilkinson divider matches both lines as they split with a rated 50ohm impedance. The Wilkinson dividers also within this circuit are resistive so through each split the power is also split between the two branches. For the delay based lossless block, they are associated with the delays of Tau_IF and Tau_LO for each respective divider circuit. The amount of each of these components is wholly determined by the number of antenna elements in the system. For example, in order to create a four branch network in Figure 9 there needs to be three Wilkinson Dividers and three transmission delays. While both the LO and IF Frequency divider circuits have the same basic tree structure, the delays are arranged in different ways. For the IF Frequency divider circuit there is a specific arrangement in which the first line has the lowest delay, and the bottom line

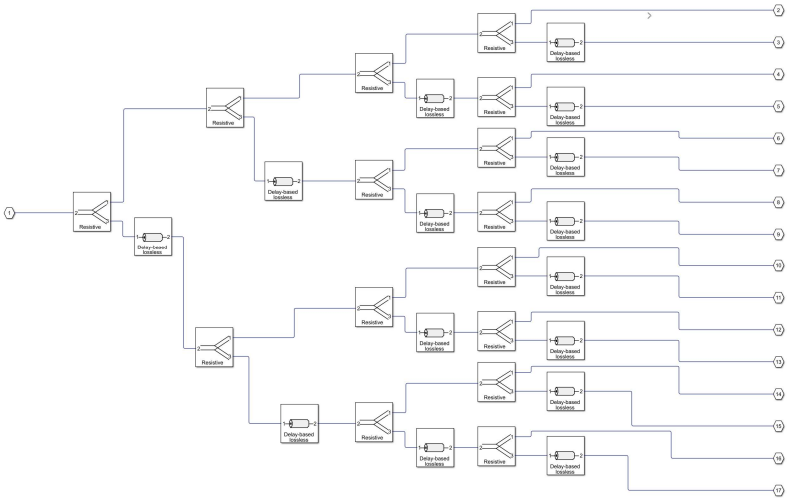
has the highest. For the LO Frequency divider, the opposite is the case, in figure 10 there highest delay occurs for the first line and the bottom line has the lowest delay for the total system.



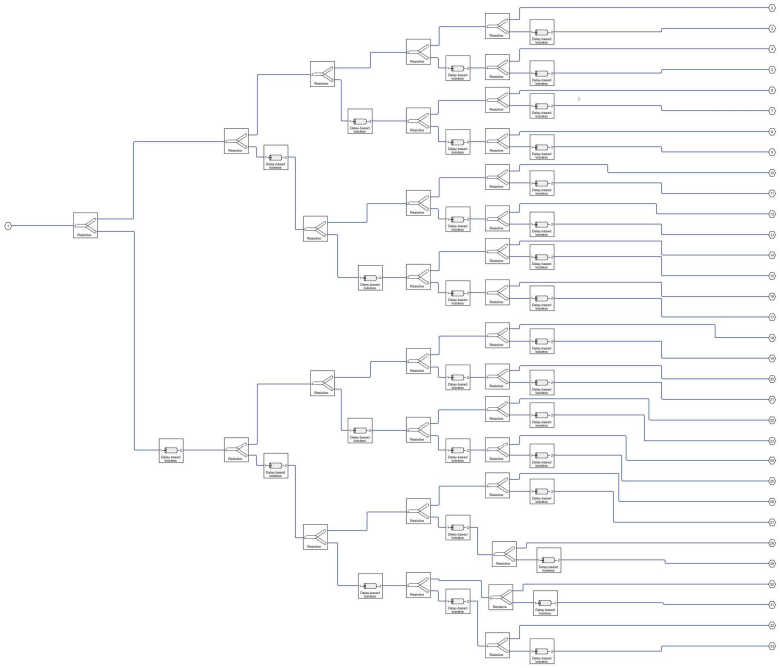
(a) Four Element Divider



(b) Eight Element Divider



(c) Sixteen Element Divider

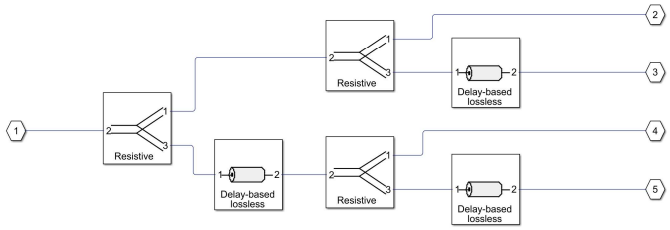


(d) Thirty-Two Element Divider

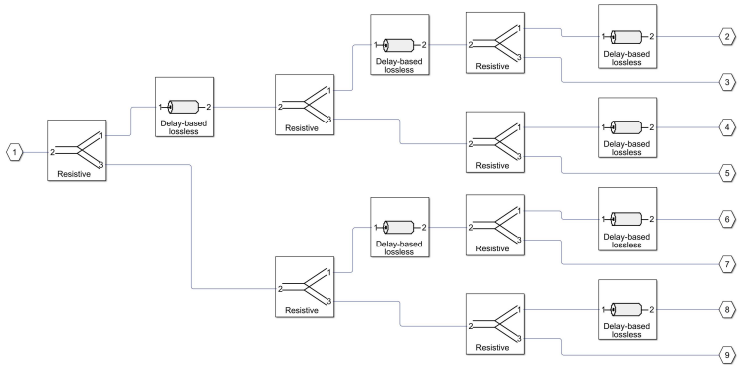
Figure 9 : IF Frequency Dividers

For the LO Frequency divider, the opposite is the case, in Figure 9 their highest delay occurs for the first line and the bottom line has the lowest delay for the total system. For both

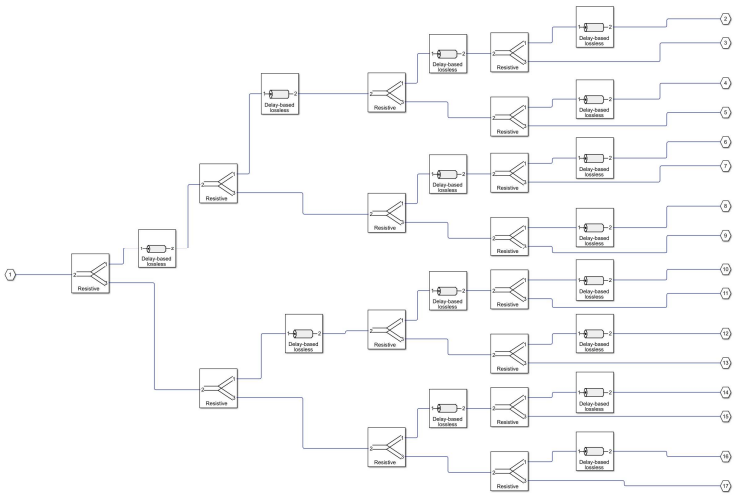
systems each delay block is also weighted differently on each branch. For example, on the Figure 10 b the first delay after the first Wilkinson divider has the delay multiplied by four. The next row of delay values subsequently is multiplied by two and the last one is specifically just the ordinary value. Depending on the size of the tree there are lessor and greater weights to each section. For the Thirty-Two element divider the first delay is specifically multiplied by sixteen.



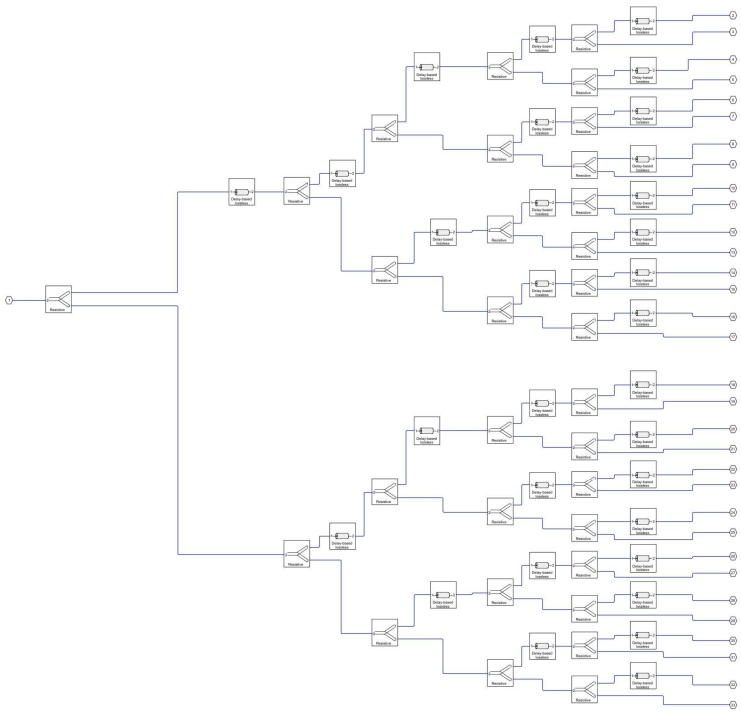
(a) Four Element Divider



(b) Eight Element Divider



(c) Sixteen Element Divider



(d) Thirty-Two Element Divider

Figure 10 : LO Frequency Dividers

Mixer and Filter Circuit

Both the LO and IF sinusoidal signals feed into the specific parts of the mixer and then fed through a Butterworth bandpass filter. The RF mixer block in Simulink does what a standard mixer does mixing both the LO and IF frequency signals. From this mixing there is also a specific variable gain of 1dB through the mixer circuit. Following the mixer block, the Butterworth bandpass filter is specifically designed to only attenuate the mixed signals such ranging from 65GHz to 155GHz. Another characteristic of the Butterworth filter is that it is an LC Tee type filter to the fifth order. Along with that, there is a passband attenuation of 3dB and it is matched at a 50 ohm source and load impedance. Each filter and mixer is numbered to how many specific lines there are in the system like in figure 11 where the four element array has four And the sixteen element array has sixteen.

Figure 11 : Four Element and Eight Element Mixer and Filter Circuit

Amplifier Block

The final part before the signal is put to analysis is its output gain through the amplifier block. This amplifier block seen in figure 12 is a specific block that has a decibel gain that can be configured towards the end of it for the signal.

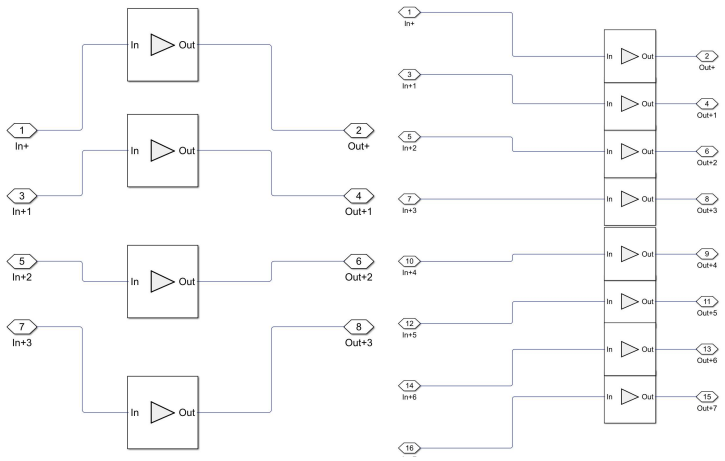


Figure 12 : Four Element and Eight Element Amplifier Circuit

These specific amplifiers will be used in the analysis to taper the amplitude using the Chebyshev weights. For the simulation that will be run there are several weights used for the multiple array systems with SLLdB tapers from 0dB, 18dB, 22dB, and 30dB. In the tables 2-5 below are the several graphs for each value of the amplifier when running the simulation.

Table 2 : Four Element Taper Weights

Antenna Number	18dB Taper Weights (dB)	20dB Taper Weights (dB)	30dB Taper Weights (dB)
1	-9.1713039	-12.60730063	-16.92503531
2	0	0	0
3	0	0	0
4	-9.1713039	-12.60730063	-16.92503531

Table 3 : Eight Element Taper Weights

Antenna Number	18dB Taper Weights (dB)	20dB Taper Weights (dB)	30dB Taper Weights (dB)
1	-7.033414486	-14.49743349	-26.77169628
2	-7.323680887	-9.282490673	-13.12677772
3	-2.361532881	-2.974185941	-4.166082366
4	0	0	0

5	0	0	0
6	-2.361532881	-2.974185941	-4.166082366
7	-7.323680887	-9.282490673	-13.12677772
8	-7.033414486	-14.49743349	-26.77169628

Table 4 : Sixteen Element Taper Weights

Antenna Number	18dB Taper Weights (dB)	20dB Taper Weights (dB)	30dB Taper Weights (dB)
1	0	-7.538893058	-24.68940511
2	-13.97858082	-15.47945199	-22.95839166
3	-10.35137618	-10.71603303	-15.71889709
4	-7.469070395	-6.975908648	-10.15806441
5	-5.244027318	-4.111701614	-5.957696324
6	-3.618078278	-2.02989122	-2.931529146
7	-2.553166292	-0.671110017	-0.967232911
8	-2.026181578	0	0
9	-2.026181578	0	0
10	-2.553166292	-0.671110017	-0.967232911
11	-3.618078278	-2.02989122	-2.931529146
12	-5.244027318	-4.111701614	-5.957696324
13	-7.469070395	-6.975908648	-10.15806441
14	-10.35137618	-10.71603303	-15.71889709
15	-13.97858082	-15.47945199	-22.95839166
16	0	-7.538893058	-24.68940511

Table 5 : Thirty-Two Element Taper Weights

Antenna Number	18dB Taper Weights (dB)	20dB Taper Weights (dB)	30dB Taper Weights (dB)
1	0	0	-16.2438586
2	-28.1545692	-21.99363594	-28.26795846
3	-26.05819987	-19.19184653	-23.84429483
4	-24.17075885	-16.69240938	-19.97281342
5	-22.47482314	-14.46318503	-16.56988535
6	-20.95636935	-12.47929532	-13.57569903
7	-19.60399981	-10.72116001	-10.94602074
8	-18.40838958	-9.173180543	-8.647317791

9	-17.36188358	-7.822831634	-6.653717801
10	-16.4581986	-6.660019585	-4.945032881
11	-15.6922004	-5.676620595	-3.505433209
12	-15.0597358	-4.866144348	-2.322534435
13	-14.55750611	-4.223487351	-1.386759486
14	-14.18297249	-3.744752541	-0.690889441
15	-13.93428649	-3.427119452	-0.229750041
16	-13.81024141	-3.268754492	0
17	-13.81024141	-3.268754492	0
18	-13.93428649	-3.427119452	-0.229750041
19	-14.18297249	-3.744752541	-0.690889441
20	-14.55750611	-4.223487351	-1.386759486
21	-15.0597358	-4.866144348	-2.322534435
22	-15.6922004	-5.676620595	-3.505433209
23	-16.4581986	-6.660019585	-4.945032881
24	-17.36188358	-7.822831634	-6.653717801
25	-18.40838958	-9.173180543	-8.647317791
26	-19.60399981	-10.72116001	-10.94602074
27	-20.95636935	-12.47929532	-13.57569903
28	-22.47482314	-14.46318503	-16.56988535
29	-24.17075885	-16.69240938	-19.97281342
30	-26.05819987	-19.19184653	-23.84429483
31	-28.1545692	-21.99363594	-28.26795846
32	0	0	-16.2438586

These amplitude tapers will be put in as weights for each specific set simulation and will be tested against the regular or 0dB taper.

4.5 Array Factor Analysis and Evaluation

The second part of this simulation is the analysis script code of the simulated antenna architecture. At the end of the Simulink simulation all the data gets sent to the workspace where it will be analyzed through the various MATLAB scripts developed.

Array Factor

Of the data collected, the first characteristic that is calculated is the Array Factor. This development is in the Main_Script_Array_Factor.m code which the phase shift of each antenna is calculated determined and then used in the Array factor equation given in (2.10). However, to get the phase shift there is a separate function developed called the line_fft.m in order to acquire the phase of the line. The line_fft.m function takes in the values outputted from the Simulink simulation, the length of the data, and sample frequency to run. Through using the Fast Fourier Transform algorithm it is possible to determine what specific frequencies this line is operating on and then then in turn extract the phase of the system. The overall output of this system is the spectrum which is the raw output from the fft, the frequency that this is occurring in, the phase of the system, the peak attenuation of the frequency, and the number of peaks in the system.

After the phases of each line are determined, these individual phase shifts are put into the Array Factor equation. Since there are four different configurations there are four different array factor equations for the number of elements in the simulation. When the element Array data is analyzed, the code will check the amount of amplitude peaks to determine the array factor equation it will go to. For example, when the eight element array data is being analyzed it have eight peaks from line_fft.m and subsequently be put in the eight element array factor equation. The amplitudes of each line will be used as data in the array factor equation if tapering is applied. These amplitudes will be normalized through dividing the given amplitudes by the largest middle amplitude. For example, in an eight antenna array system the first amplitude will be normalized by the amplitude of line four since the fourth one is the lowest taper. These amplitude weights will be multiplied at each element of the array factor to produce the tapering within the system.

All the Array Factor data will be plotted in several forms. The first will be a regular cartesian plot which will show the Array Factor pattern in decibels over the shift from -pi/2 to pi/2. There will also be a polar pattern plot that will instead be in magnitude instead of decibels. Lastly, there will be 3d plots of each of these array factors over a frequency shift from 25 GHz to 35 GHz and an angle shift from -pi/2 to pi/2.

Steering Angle

Another crucial benchmark for the frequency scanned array evaluation is how accurate the values are to the theoretical calculations of the steering angle. In the file Steering_Angle.m theoretical equation for the steering angle of the system is used and plotted against the simulated values and the derivative at the center frequency. There are twenty-one specific simulated data points from input frequencies from 25GHz to 35GHz. The specific derivative value at the center frequency is relevant to this simulation due to the fact that the derivative at the center frequency is a standardizing benchmark for the system. There will be four of these plots for each different array and a function determining the accuracy of the simulation relative to the steering angle values.

Beamwidth

The beamwidth of the system was the last characteristic in need of measurement for the frequency scanned array system. This specific simulation compares the beamwidth estimation equation against the actual beamwidth derived from the system. To get the beamwidth empirically from the array factor graph there was a Beamwidth.m function developed to measure from the main lobe value minus 3dB. At 3dB the position would from the lobe would be found

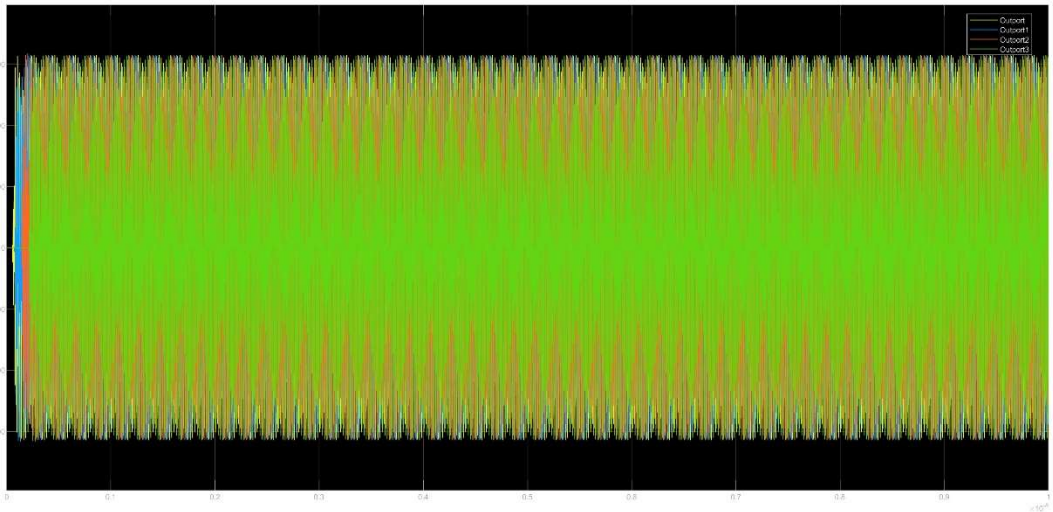
and the distance from in degrees from that point to the other 3dB point on the other side of the symmetrical beampattern will be measured. These beamwidth measurements would be collected and plotted in several graphs against the equation.

Chapter 5

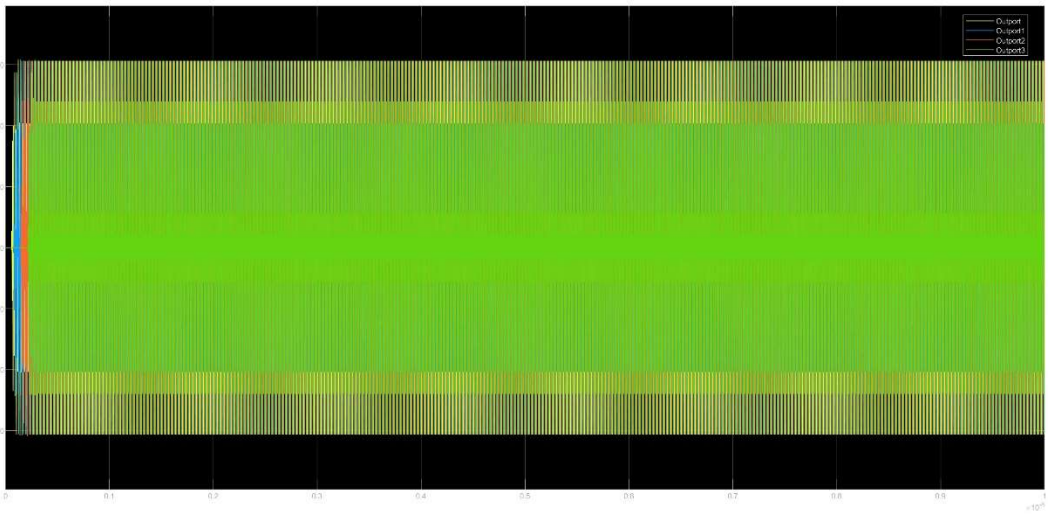
Simulation Results

5.1 Simulink Analysis

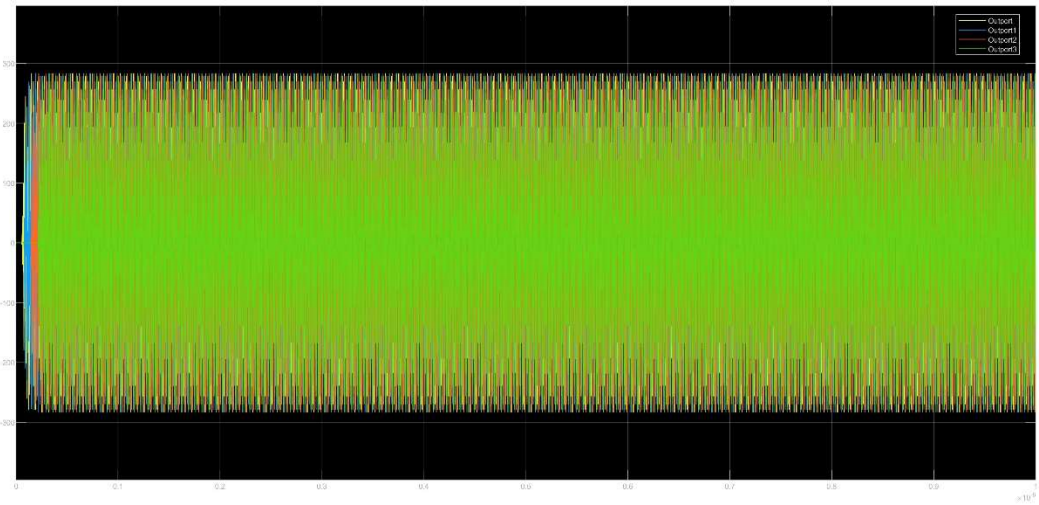
The first results from this model where the scope output simulations showing each voltage output and the frequency shifts from the delays. In Figure 13 there for the four element array there is a noticeable shift between each of the 4 element outputs for the 25GHz input frequency. In (b) the 30GHz frequency has every part practically lined up in unison. Also, the frequency of the LO and IF are properly mixed.



(a) 25GHz Input Frequency



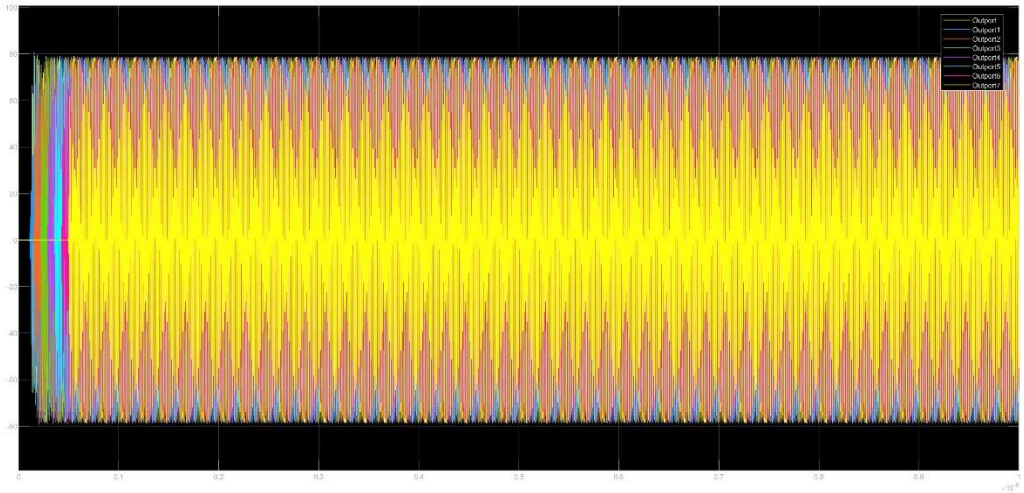
(b) 30GHz Input Frequency



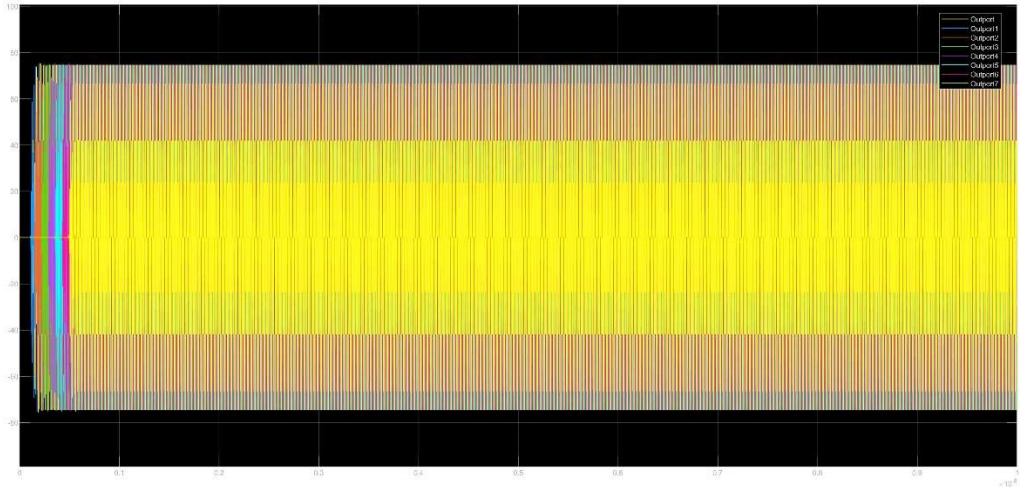
(c) 35GHz Input Frequency

Figure 13 : 4 Element Simulation Output

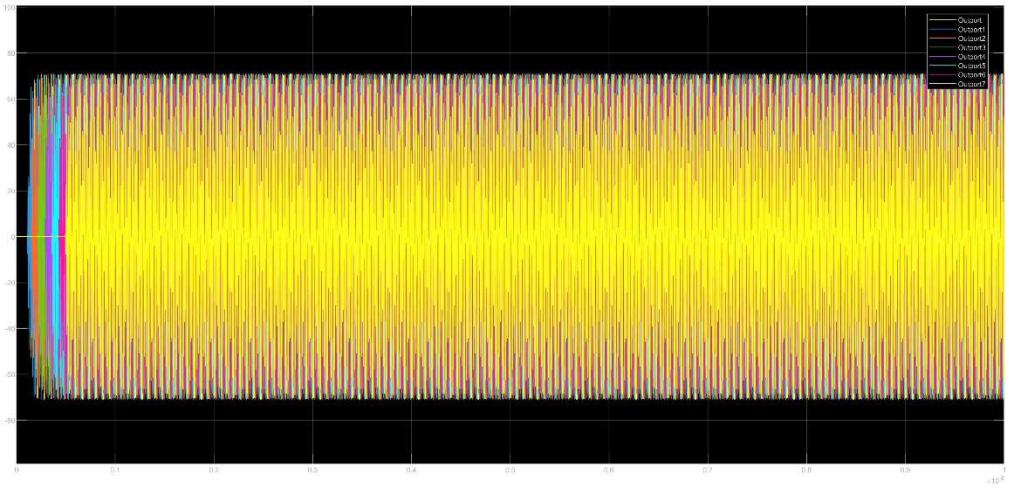
For Figure 14 the same cam be said when increasing the number of elements. Each individual shift is noticeable and there is a clear differentiation from where each delay starts and ends form the highest to the lowest delay. Each one is evenly spaced at 25GHz and 35GHz and each one is particularly aligned for the 30GHz section in (b).



(a) 25GHz Input Frequency



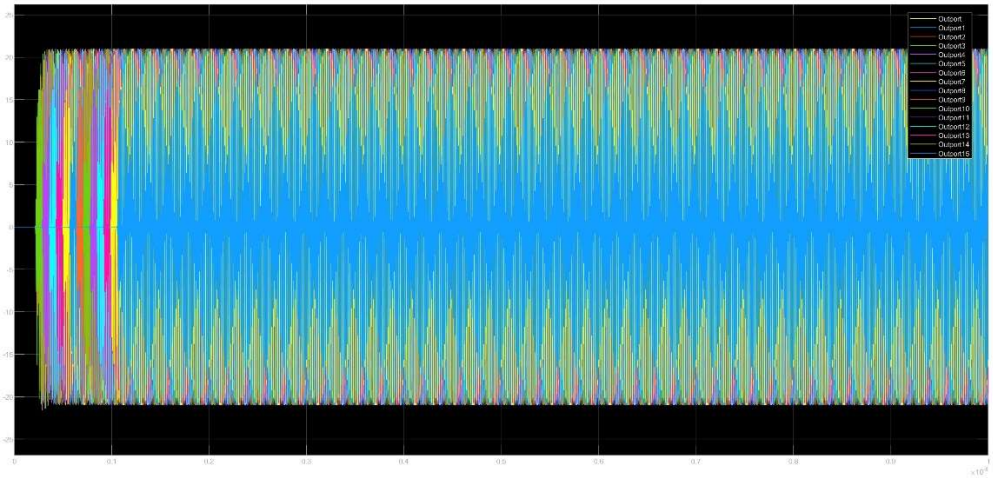
(b) 30GHz Input Frequency



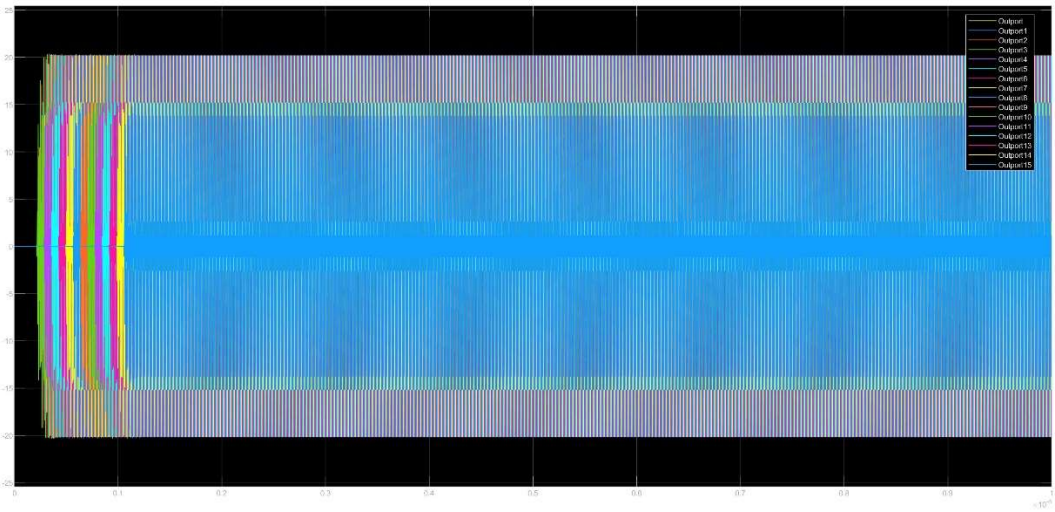
(c) 35GHz Input Frequency

Figure 14 : 8 Element Simulation Output

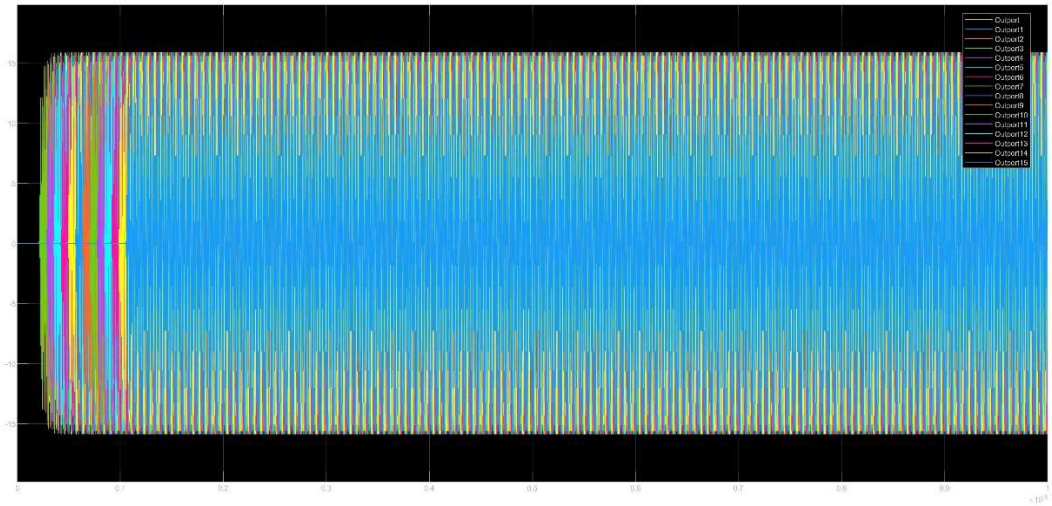
The pattern I discussed before continues in the below figure s 15 and 16 where each of the LO and IF frequencies are mixed and correspond to the proper phase shifts at each specific frequency.



(a) 25GInput Frequency

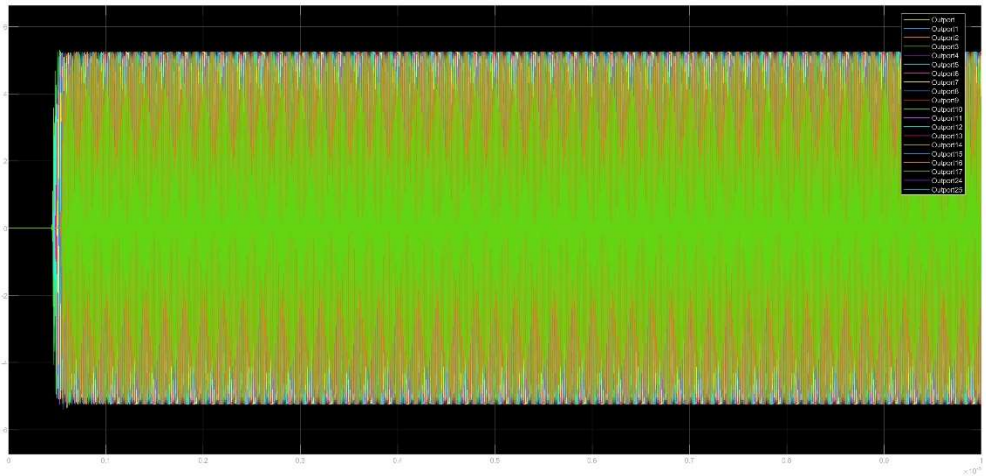


(b) 30GHz Input Frequency

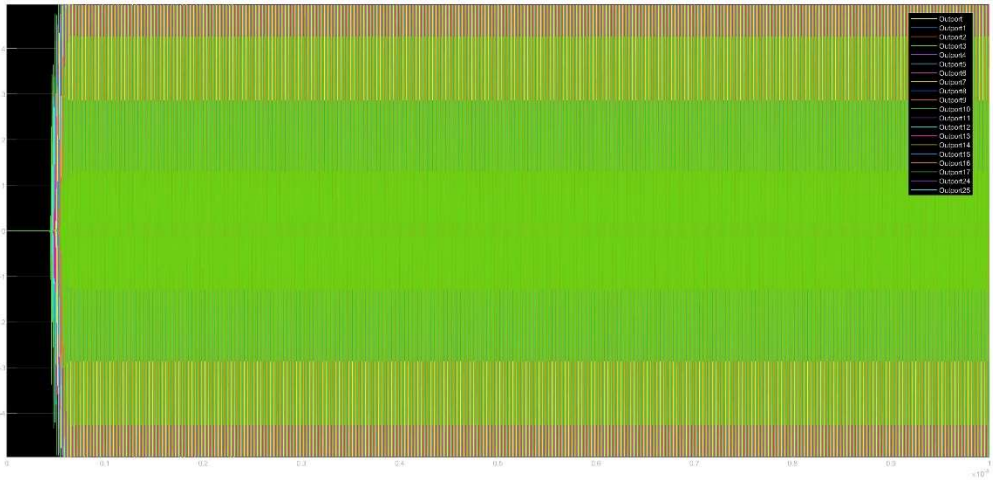


(c) 35GHz Input Frequency

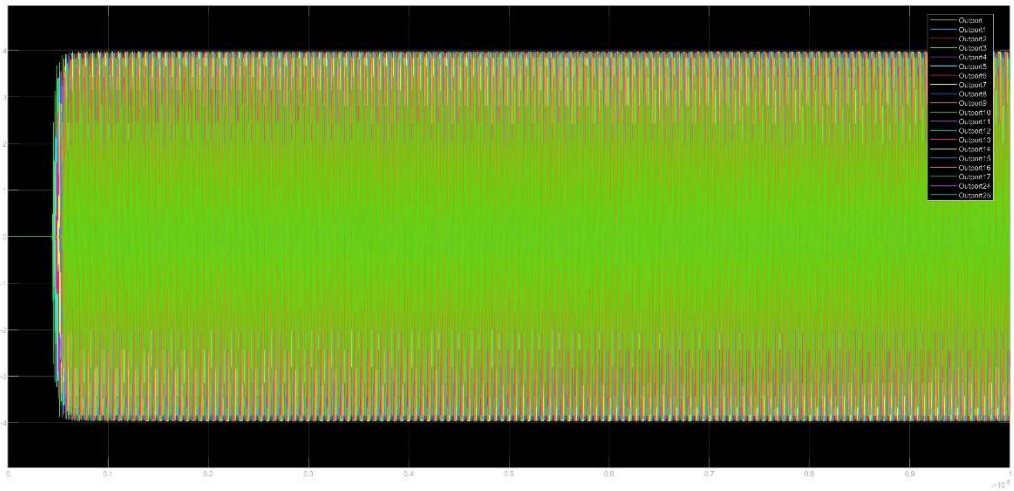
Figure 15 : 16 Element Simulation Output



(a) 25GHz Input Frequency



(b) 30GHz Input Frequency



(c) 35GHz Input Frequency

Figure 16 : 32 Element Simulation Output

Overall, the data from the scope is a match with what was expected from each delay shift. The 30GHz IF frequency had every specific part aligned with the no phase shift. The 25GHz and 35GHz had the -45 to 45 degrees shift respectively. As a final test in Figure 17, the tapering effect of the amplifiers is brought on the scope output.

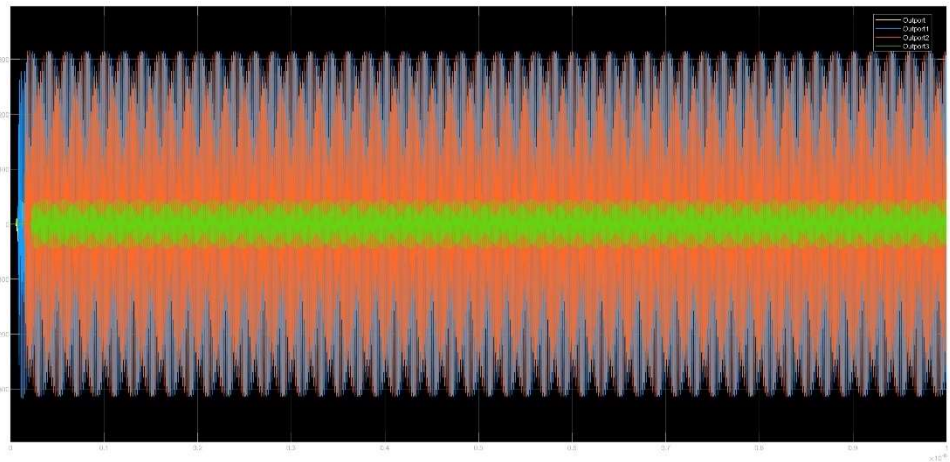
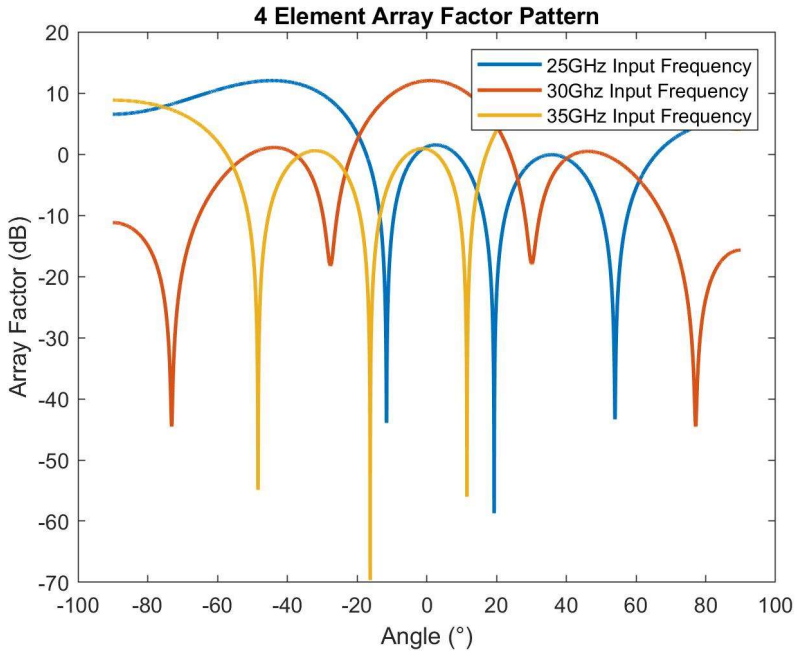


Figure 17: 4 Element Array 30dB tapering

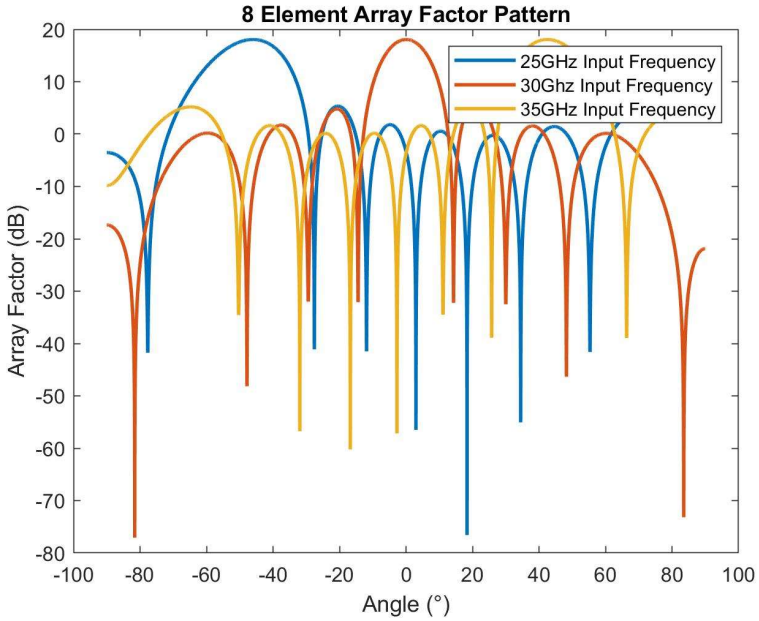
As demonstrated in Figure 17 antenna elements 1 and 4 or outport and outport 3 are specifically are tapered at 30dB with the amplifier. While outport 1 and 2 are not tapered and remain at the same value.

5.2 Array Factor Results

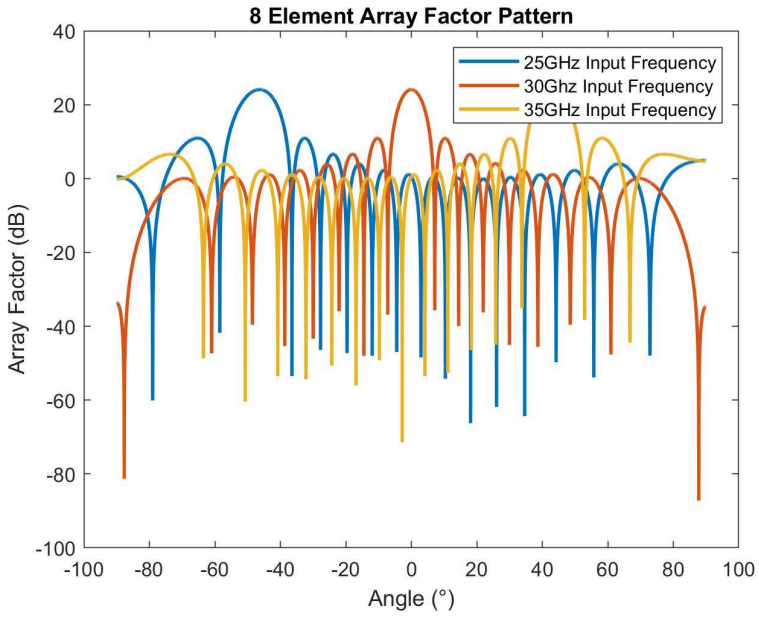
The overall array factor plots have been the most important evaluation of each of these specific arrays. Each individual pattern will demonstrate the intensity and place which this beam is shifting and at what values. In figure 18 (a) the four element array has a broad beamwidth and sidelobes with its shifts. Each figure within figure 18 hold the shifts to the respective areas on the plot. 25GHz is at the -45 degrees and 35GHz is found to have its main lobe at 45degrees. While 30GHz input has a main lobe found at 0 degrees. All of this is specifically expected in the design of this system. Especially the fact that as the number of elements increase the main lobe becomes more attenuated and at a less wide beamwidth. Also, it is observed that the side lobe suppression is less at increased elements in the array. A comparison from (a) to (d) in Figure 18 shows this specific relation.



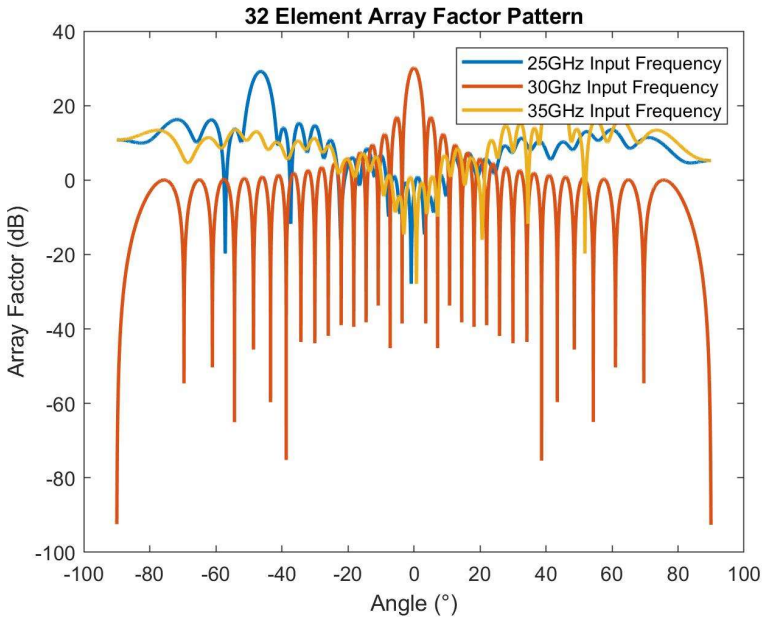
(a) 4 Element



(b) 8 Element


8 Element Array Factor Pattern
25GHz Input Frequency
30Ghz Input Frequency
35GHz Input Frequency
Array Factor (dB)
Angle (°)
40
20
0
-20
-40
-60
-80
-100
-100
-80
-60
-40
-20
0
20
40
60
80
100

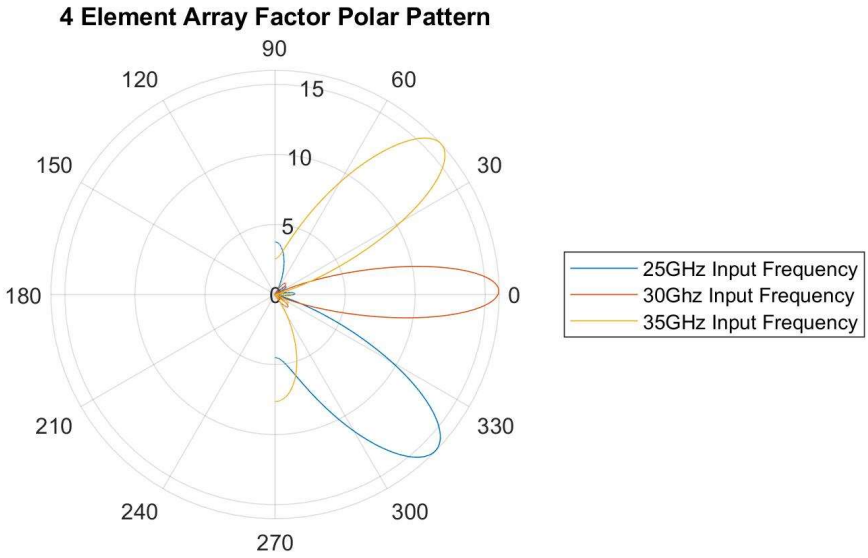
(c) 16 Element



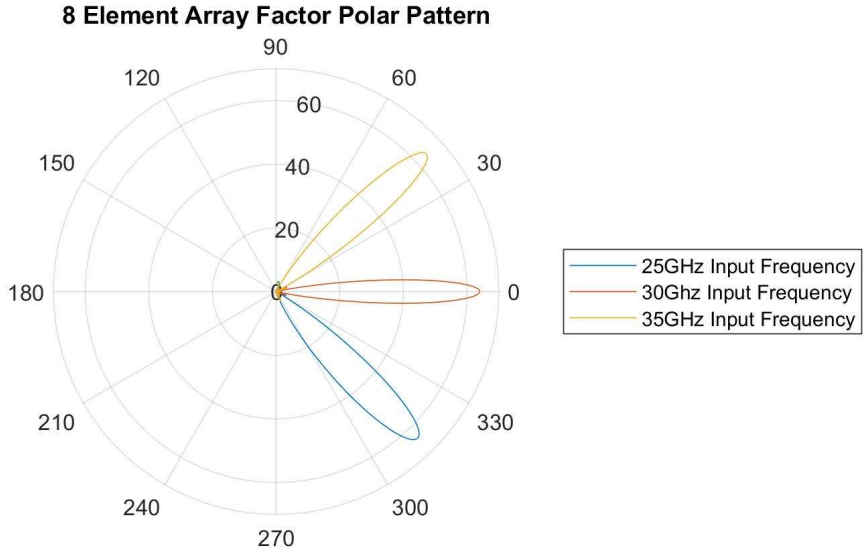
(d) 32 Element

Figure 18 : Array Factor Plot Shift

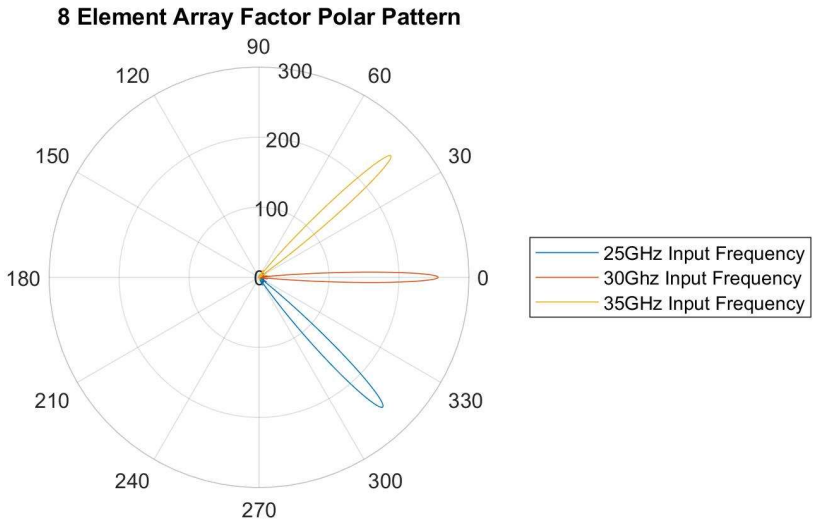
Within Figure 19 the polar plots of each of these array factors the difference is even more stark as there is a more narrow beam pattern and decrease in side lobes as the elements increase.



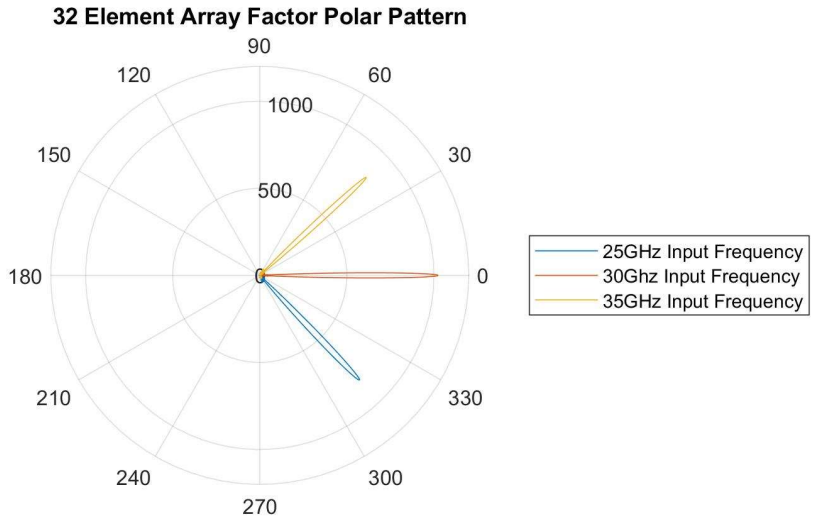
(a) 4 Element



(b) 8 Element



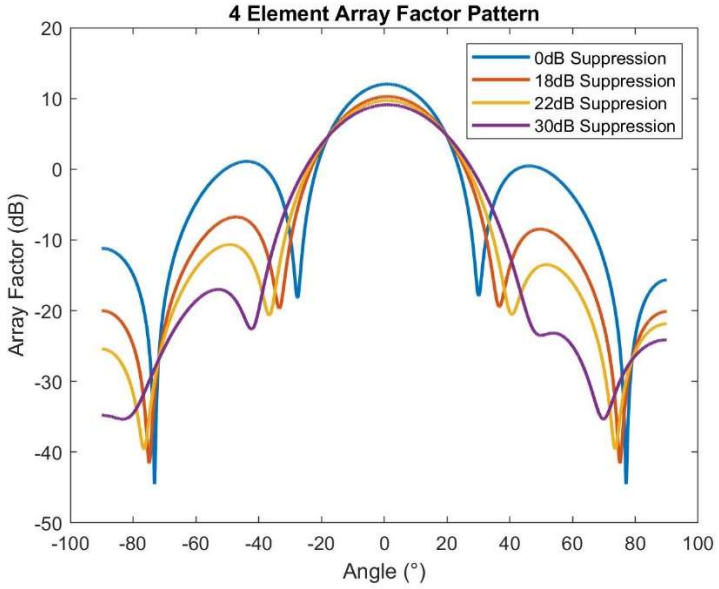
(c) 16 Element



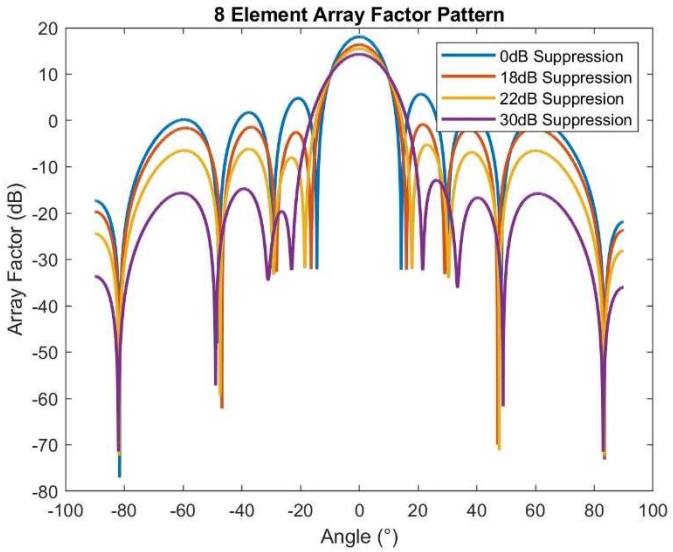
(d) 32 Element

Figure 19 : Array Factor Shift Polar Plot

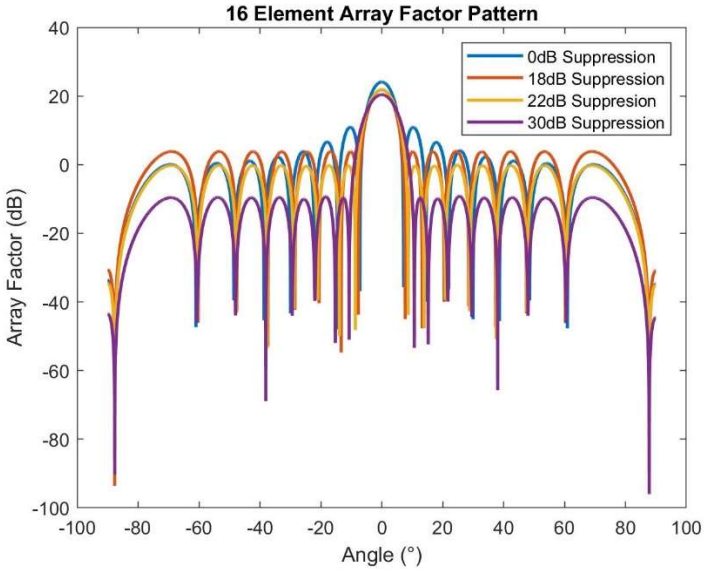
The tapering Chebyshev weights are also working in accordance to the specific SLL suppression for the cartesian array factor plots and the polar factor plots. Each side lobe is increasing minimized at a normalized center input frequency of 30GHz.



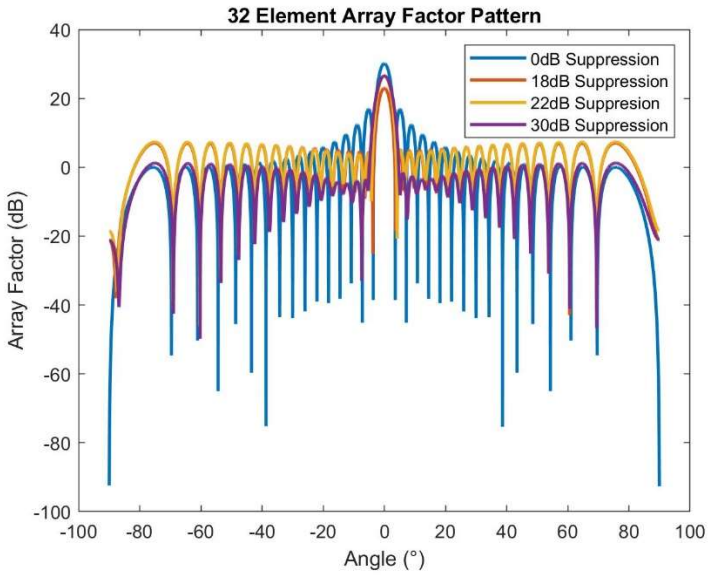
(a) 4 Element



(b) 8 Element



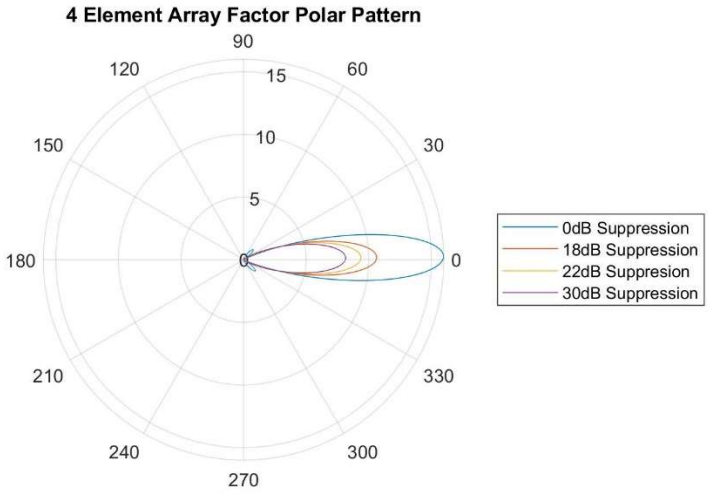
(c) 16 Element



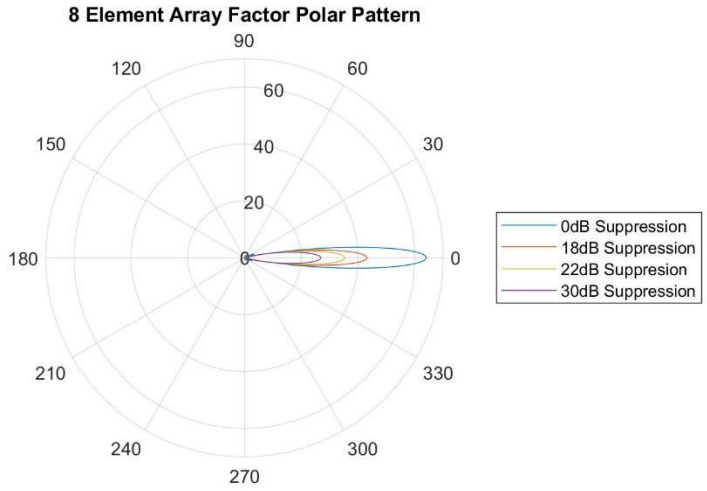
(d) 32 Element

Figure 20 : Array Factor Tapering Plot

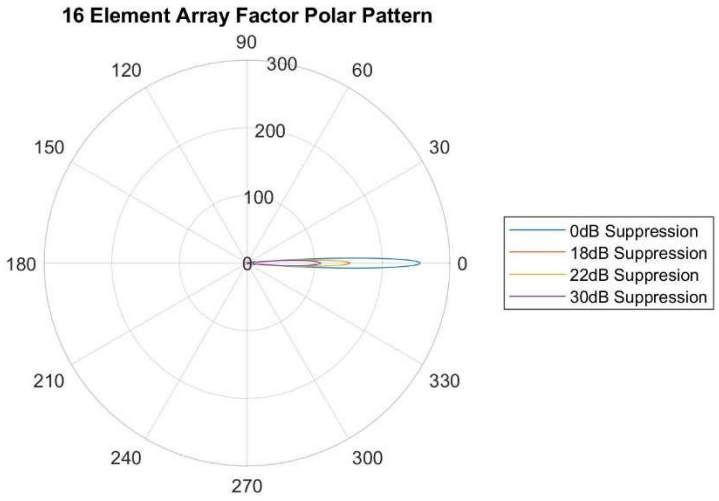
Each of the tapers become more pronounced and specifics as the system elements increase. At (d) each element taper is relatively uniform and brought to a greater and greater suppression as opposed to the lobes in (a). For all of these lobes there is a decrease within the main lobe attenuation. This main lobe attenuation decrease is more pronounced with the increase in the number of elements and within the polar plots of Figure 21.



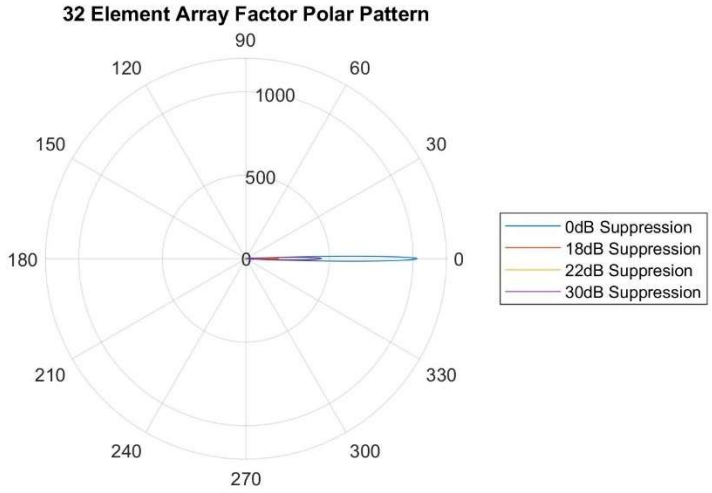
(a) 4 Element



(b) 8 Element



(c) 16 Element

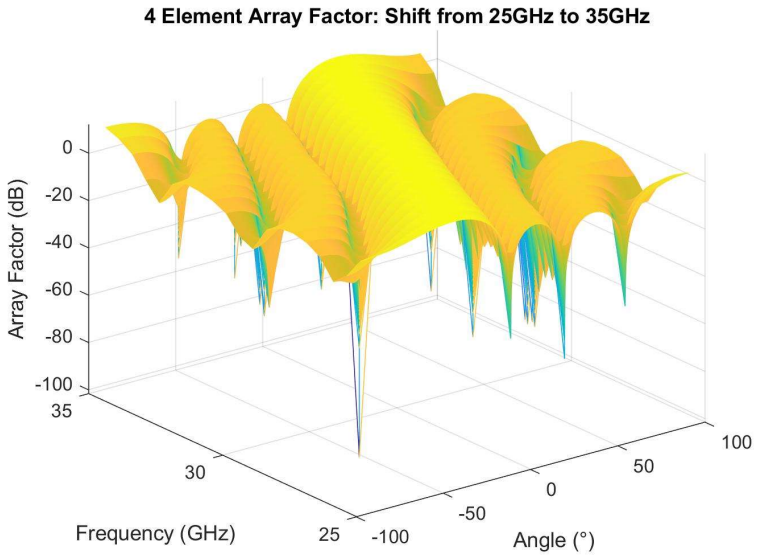


(d) 32 Element

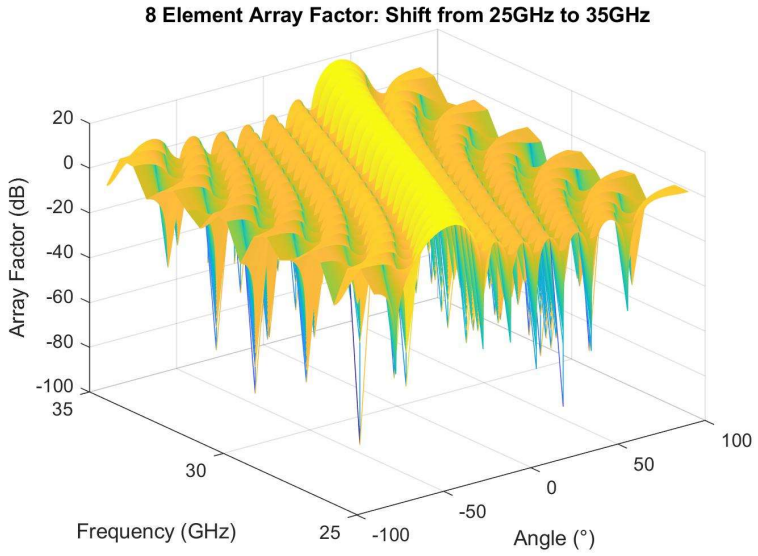
Figure 21 : Array Factor Tapering Polar Plot

Overall, the difference between the main lobe and sidelobes across each frequency can be seen the following 3d plots in figure 22. There is a coordinated shift with each angle and frequency. From

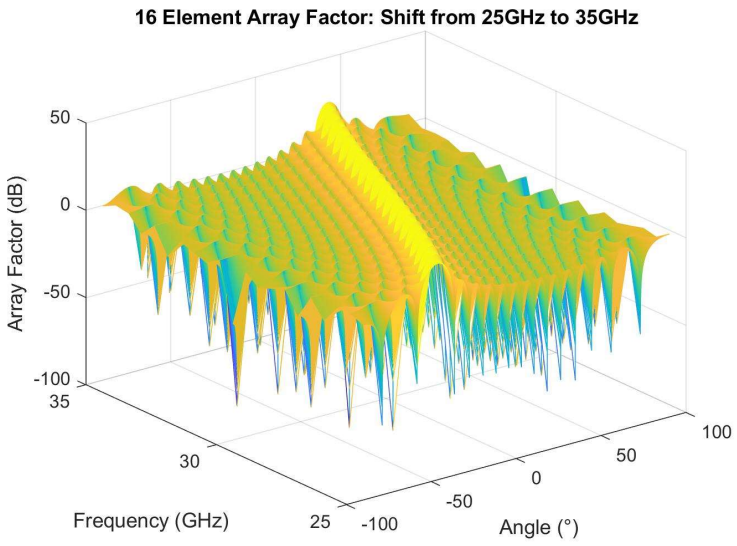
(a) to (d) the contours of the graphs show the sidelobes and the main lobe thinning out with a greater number of elements.



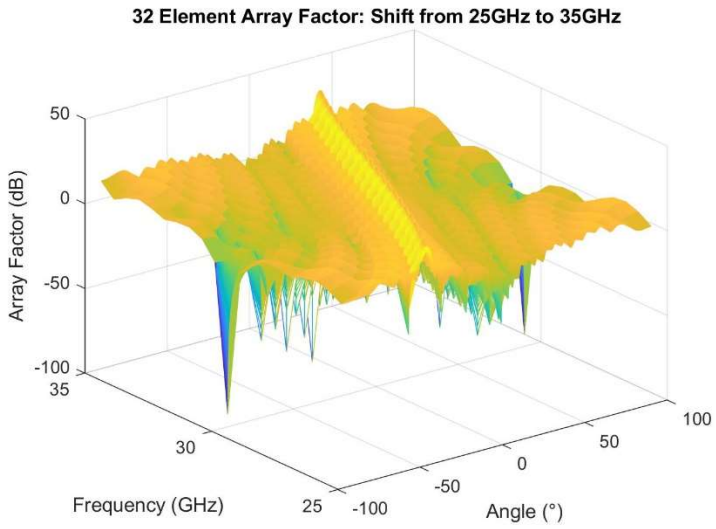
(a) 4 Element



(b) 8 Element



(c) 16 Element



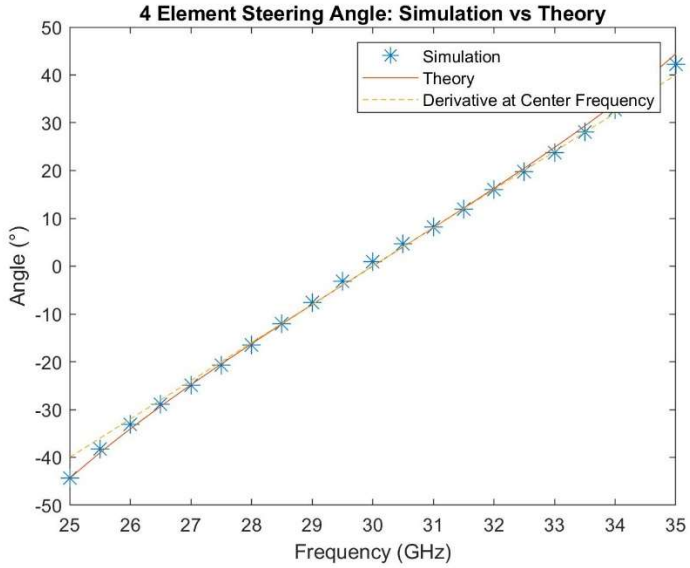
(d) 32 Element

Figure 22 : 3D Plot of Array Factor

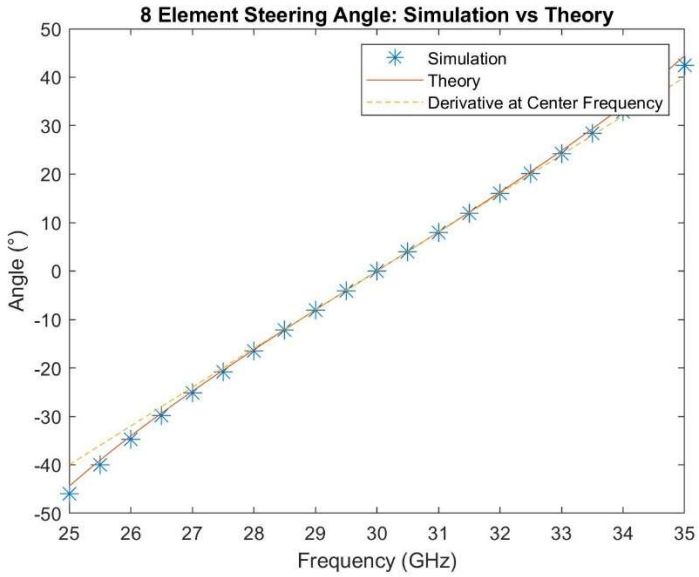
The array factor of each antenna array element system is very much in line with the data brought forth in the theory. All of the taper weights are attenuating and performing better with a greater number of elements. The shifts at each specific frequency have also been at each standard position.

5.3 Beam Steering Results

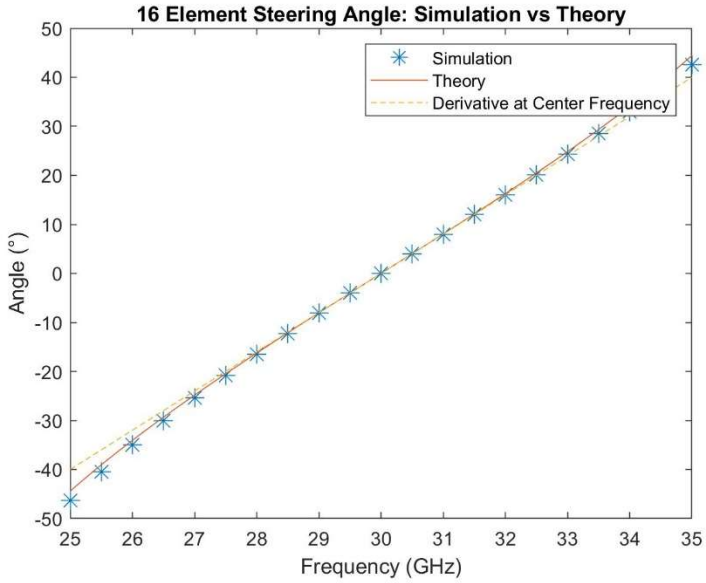
The beam steering angle's direction within antenna array is the most important factor of this design. Accurate steering is absolutely necessary in order for mmWave applications and communication. Within the following figure 23 set there will be a comparison between the antenna array data from the Simulink model and the calculated steering angle. Along with that is the specific derivative at the center frequency for analysis.



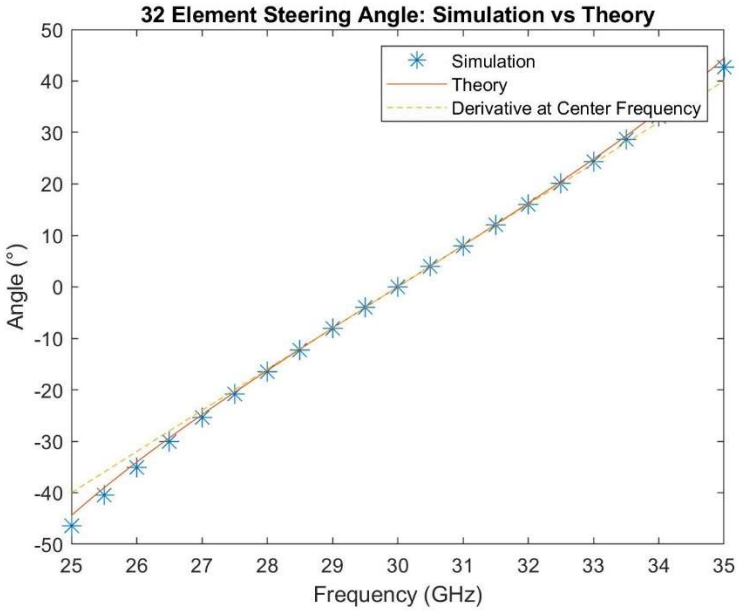
(a) 4 Element



(b) 8 Element



(c) 16 Element



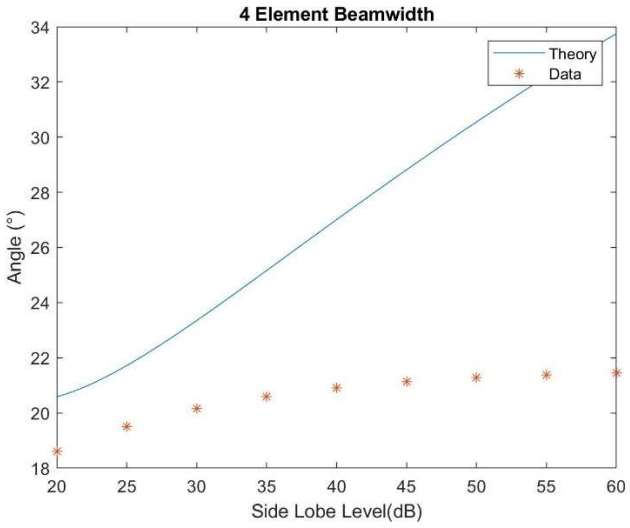
(d) 32 Element

Figure 23 : Steering Angle Simulation vs Theory

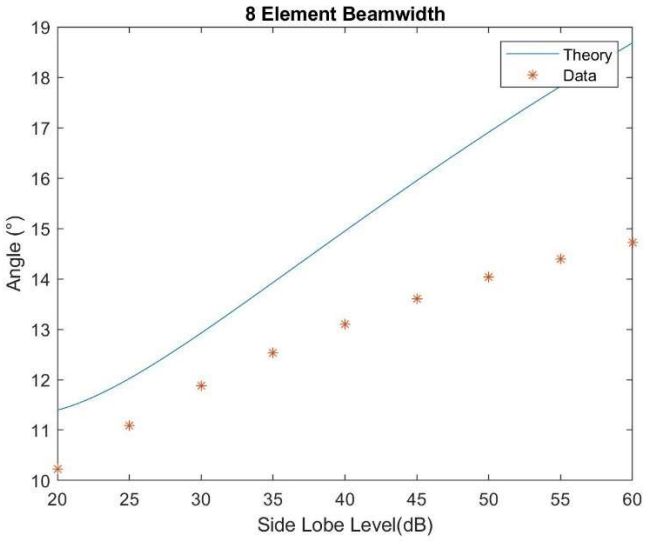
From (a)-(d) in the following graphs there is a great match between the simulation data of the steering angle position and the theoretical values form 25GHz to 35GHz. Each specific element array was varying only slightly on each specific point veering of mostly towards 25GHz and 35GHz around the ends of the simulation. Overall, from the graphs comparing the beam steering angle across all of these graphs the theory and data match with a high degree of accuracy.

5.4 Beamwidth Results

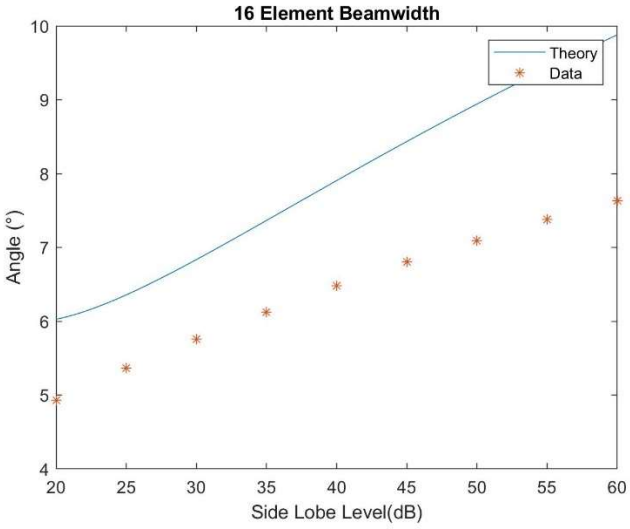
The beamwidth results for each specific antenna element design is an important predictive value for the Chebyshev array. With each successive beamwidth pattern there is a widening of the main lobe within the system. As a result it is important to maintain track of this and compare it with the theoretical values.



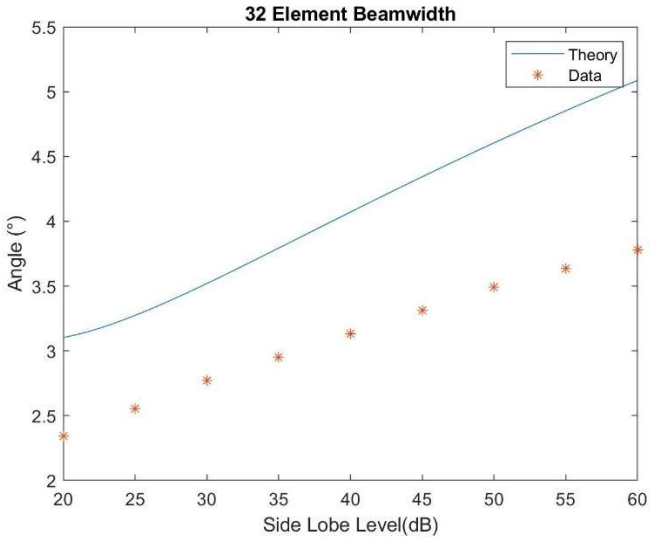
(a) 4 Element



(b) 8 Element



(c) 16 Element



(d) 32 Element

Figure 24 : Simulated vs Calculated Beamwidth

From figure 24 from (a) to (d) there is a great deviation from each specific theoretical value and the data collected. The patterns within these graphs tend to be the same however each of these beam widths are tapered lower. The (a) 4 element arrays suffered from the most disparity between theory and simulation as the taper specifically follows a much different nonlinear pattern of beamwidth. Not to mention the disparity between each of the specific values. For the rest of the plots this continues to ring true but tend not to deviate as wildly as the general trend of the data follows the theoretical calculation.

Chapter 6

Conclusion

Overall, a working simulation of a mmWave frequency scanned array antenna in MATLAB has been developed. Improving upon the prior Simulink Model and MALTAB scripts, I was able to get accurate beam steering from the device from 145GHz to 155GHz. I have been able to then tests the impact that the number of elements in the simulation has on its overall performance. Along with that, SLL tapering has been another benchmark for the system using a weighted tapering amplifier in the MATLAB main script. Also, the beamwidth values have been another exclusive benchmark determining the specific pattern of the system and its variability.

For future research in the development of a frequency scanned array model reference will be made to new novel frequency scanned array antenna designs to complement the architecture and further give an in depth simulation. In [11] leaky wave antenna designs have been used for high frequency devices such as this and combined with the delay imposed by the various methods and designs. This can specifically be modeled within the antenna array toolbox system in Simulink. The antenna system will specifically have the resources to design and implement these unique models of antenna arrays and match it specifically to the system.

Appendix A

MATLAB Script

Main_Script_Array_Factor.m

```
start = 2000000;

stop = 6250001;

time = out.tout(start:stop,1);

data_length = length(time);

%start = 2000000; works for 32 element at 30 and 25ghz step size

%0.160e-7micro

%stop = 6250001




%1000001:9000000,1 works for everything step size 0.1*10^-7 microseconds


line_1 = out.line_1(start:stop,1);

max_line_1 = max(line_1);

line_2 = out.line_2(start:stop,1);

max_line_2 = max(line_2);

line_3 = out.line_3(start:stop,1);

max_line_3 = max(line_3);

line_4 = out.line_4(start:stop,1);

max_line_4 = max(line_4);
```

```
line_5 = out.line_5(start:stop,1);
max_line_5 = max(line_5);
line_6 = out.line_6(start:stop,1);
max_line_6 = max(line_6);
line_7 = out.line_7(start:stop,1);
max_line_7 = max(line_7);
line_8 = out.line_8(start:stop,1);
max_line_8 = max(line_8);
line_9 = out.line_9(start:stop,1);
max_line_9 = max(line_9);
line_10 = out.line_10(start:stop,1);
max_line_10 = max(line_10);
line_11 = out.line_11(start:stop,1);
max_line_11 = max(line_11);
line_12 = out.line_12(start:stop,1);
max_line_12 = max(line_12);
line_13 = out.line_13(start:stop,1);
max_line_13 = max(line_13);
line_14 = out.line_14(start:stop,1);
max_line_14 = max(line_14);
line_15 = out.line_15(start:stop,1);
max_line_15 = max(line_15);
line_16 = out.line_16(start:stop,1);
```

```
max_line_16 = max(line_16);
line_17 = out.line_17(start:stop,1);
max_line_17 = max(line_17);
line_18 = out.line_18(start:stop,1);
max_line_18 = max(line_18);
line_19 = out.line_19(start:stop,1);
max_line_19 = max(line_19);
line_20 = out.line_20(start:stop,1);
max_line_20 = max(line_20);
line_21 = out.line_21(start:stop,1);
max_line_21 = max(line_21);
line_22 = out.line_22(start:stop,1);
max_line_22 = max(line_22);
line_23 = out.line_23(start:stop,1);
max_line_23 = max(line_23);
line_24 = out.line_24(start:stop,1);
max_line_24 = max(line_24);
line_25 = out.line_25(start:stop,1);
max_line_25 = max(line_25);
line_26 = out.line_26(start:stop,1);
max_line_26 = max(line_26);
line_27 = out.line_27(start:stop,1);
max_line_27 = max(line_27);
```

```
line_28 = out.line_28(start:stop,1);
max_line_28 = max(line_28);
line_29 = out.line_29(start:stop,1);
max_line_29 = max(line_29);
line_30 = out.line_30(start:stop,1);
max_line_30 = max(line_30);
line_31 = out.line_31(start:stop,1);
max_line_31 = max(line_31);
line_32 = out.line_32(start:stop,1);
max_line_32 = max(line_32);

fs = 1/(out.tout(2,1)-out.tout(1,1));

[y1, f1, phi_1, f_peak_1, num_of_peaks_1] = line_fft(line_1, data_length, fs);
[y2, f2, phi_2, f_peak_2, num_of_peaks_2] = line_fft(line_2, data_length, fs);
[y3, f3, phi_3, f_peak_3, num_of_peaks_3] = line_fft(line_3, data_length, fs);
[y4, f4, phi_4, f_peak_4, num_of_peaks_4] = line_fft(line_4, data_length, fs);
[y5, f5, phi_5, f_peak_5, num_of_peaks_5] = line_fft(line_5, data_length, fs);
[y6, f6, phi_6, f_peak_6, num_of_peaks_6] = line_fft(line_6, data_length, fs);
[y7, f7, phi_7, f_peak_7, num_of_peaks_7] = line_fft(line_7, data_length, fs);
[y8, f8, phi_8, f_peak_8, num_of_peaks_8] = line_fft(line_8, data_length, fs);
[y9, f9, phi_9, f_peak_9, num_of_peaks_9] = line_fft(line_9, data_length, fs);
[y10, f10, phi_10, f_peak_10, num_of_peaks_10] = line_fft(line_10, data_length, fs);
```

```
[y11, f11, phi_11, f_peak_11, num_of_peaks_11] = line_fft(line_11, data_length, fs);
[y12, f12, phi_12, f_peak_12, num_of_peaks_12] = line_fft(line_12, data_length, fs);
[y13, f13, phi_13, f_peak_13, num_of_peaks_13] = line_fft(line_13, data_length, fs);
[y14, f14, phi_14, f_peak_14, num_of_peaks_14] = line_fft(line_14, data_length, fs);
[y15, f15, phi_15, f_peak_15, num_of_peaks_15] = line_fft(line_15, data_length, fs);
[y16, f16, phi_16, f_peak_16, num_of_peaks_16] = line_fft(line_16, data_length, fs);
[y17, f17, phi_17, f_peak_17, num_of_peaks_17] = line_fft(line_17, data_length, fs);
[y18, f18, phi_18, f_peak_18, num_of_peaks_18] = line_fft(line_18, data_length, fs);
[y19, f19, phi_19, f_peak_19, num_of_peaks_19] = line_fft(line_19, data_length, fs);
[y20, f20, phi_20, f_peak_20, num_of_peaks_20] = line_fft(line_20, data_length, fs);
[y21, f21, phi_21, f_peak_21, num_of_peaks_21] = line_fft(line_21, data_length, fs);
[y22, f22, phi_22, f_peak_22, num_of_peaks_22] = line_fft(line_22, data_length, fs);
[y23, f23, phi_23, f_peak_23, num_of_peaks_23] = line_fft(line_23, data_length, fs);
[y24, f24, phi_24, f_peak_24, num_of_peaks_24] = line_fft(line_24, data_length, fs);
[y25, f25, phi_25, f_peak_25, num_of_peaks_25] = line_fft(line_25, data_length, fs);
[y26, f26, phi_26, f_peak_26, num_of_peaks_26] = line_fft(line_26, data_length, fs);
[y27, f27, phi_27, f_peak_27, num_of_peaks_27] = line_fft(line_27, data_length, fs);
[y28, f28, phi_28, f_peak_28, num_of_peaks_28] = line_fft(line_28, data_length, fs);
[y29, f29, phi_29, f_peak_29, num_of_peaks_29] = line_fft(line_29, data_length, fs);
[y30, f30, phi_30, f_peak_30, num_of_peaks_30] = line_fft(line_30, data_length, fs);
[y31, f31, phi_31, f_peak_31, num_of_peaks_31] = line_fft(line_31, data_length, fs);
[y32, f32, phi_32, f_peak_32, num_of_peaks_32] = line_fft(line_32, data_length, fs);
```

```
peaks_check =
num_of_peaks_1+num_of_peaks_2+num_of_peaks_3+num_of_peaks_4+num_of_peaks_5+nu
m_of_peaks_6+num_of_peaks_7+num_of_peaks_8+num_of_peaks_9+num_of_peaks_10+num
_of_peaks_11+num_of_peaks_12+num_of_peaks_13+num_of_peaks_14+num_of_peaks_15+nu
m_of_peaks_16+num_of_peaks_17+num_of_peaks_18+num_of_peaks_19+num_of_peaks_20+
num_of_peaks_21+num_of_peaks_22+num_of_peaks_23+num_of_peaks_24+num_of_peaks_2
5+num_of_peaks_26+num_of_peaks_27+num_of_peaks_28+num_of_peaks_29+num_of_peaks
_30+num_of_peaks_31+num_of_peaks_32

Freq = linspace(25,35,10);

%figure(1)
%plot(f7,y7);
if (peaks_check == 8)
    theta = linspace(-pi/2,pi/2,10001);
    c = 3e8;
    f1 = f_peak_1(1);
    f2 = f_peak_1(2);
    lamda_1 = c/f1;
    lamda_2 = c/f2;
    lamda = c/(150e9);
    d = lamda/2;
```

```
phi_1_1 = phi_1(1)+0*pi/lamda_1*sin(theta)*d;
phi_2_1 = phi_2(1)+2*pi/lamda_1*sin(theta)*d;
phi_3_1 = phi_3(1)+4*pi/lamda_1*sin(theta)*d;
phi_4_1 = phi_4(1)+6*pi/lamda_1*sin(theta)*d;

phi_1_2 = phi_1(2)+0*pi/lamda_2*sin(theta)*d;
phi_2_2 = phi_2(2)+2*pi/lamda_2*sin(theta)*d;
phi_3_2 = phi_3(2)+4*pi/lamda_2*sin(theta)*d;
phi_4_2 = phi_4(2)+6*pi/lamda_2*sin(theta)*d;

Phi_4_all = [phi_1(2),phi_2(2),phi_3(2),phi_4(2)]

new1 = chebwin(4,18);
A1 = (abs(exp(1i*phi_1_2)+exp(1i*phi_2_2)+exp(1i*phi_3_2)+exp(1i*phi_4_2))).^2;
A2 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2))).^2;
new1 = chebwin(4,22);
A3 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2))).^2;
new1 = chebwin(4,30);
```

```
A4 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2))).^2;

figure(4)
polarplot(theta, A1);
hold on
%polarplot(theta, A2);
%polarplot(theta, A3);
%polarplot(theta, A4);
title("4 Element Array Factor Polar Pattern")
legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")
figure(5)
plot(theta/pi*180, 10*log10(A1), 'LineWidth',1.7);
hold on
%plot(theta/pi*180, 10*log10(A2), 'LineWidth',1.7);
%plot(theta/pi*180, 10*log10(A3), 'LineWidth',1.7);
%plot(theta/pi*180, 10*log10(A4), 'LineWidth',1.7);
title("4 Element Array Factor Pattern")
ylabel("Array Factor (dB)")
xlabel("Angle (°)")
legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")
```

```
else if (peaks_check == 32)

    theta = linspace(-pi/2,pi/2,10001);
    c = 3e8;
    f1 = f_peak_1(1);
    f2 = f_peak_1(2);
    lamda_1 = c/f1;
    lamda_2 = c/f2;
    lamda = c/(150e9);
    d = lamda/2;

    phi_16_all =
[phi_1(2),phi_2(2),phi_3(2),phi_4(2),phi_5(2),phi_6(2),phi_7(2),phi_8(2),phi_9(2),phi_10(2),ph
i_11(2),phi_12(2),phi_13(2),phi_14(2),phi_15(2),phi_16(2)]

    phi_1_1 = phi_1(1)+0*pi/lamda_1*sin(theta)*d;
    phi_2_1 = phi_2(1)+2*pi/lamda_1*sin(theta)*d;
    phi_3_1 = phi_3(1)+4*pi/lamda_1*sin(theta)*d;
    phi_4_1 = phi_4(1)+6*pi/lamda_1*sin(theta)*d;
    phi_5_1 = phi_5(1)+8*pi/lamda_1*sin(theta)*d;
    phi_6_1 = phi_6(1)+10*pi/lamda_1*sin(theta)*d;
    phi_7_1 = phi_7(1)+12*pi/lamda_1*sin(theta)*d;
```

```
phi_8_1 = phi_8(1)+14*pi/lamda_1*sin(theta)*d;

phi_9_1 = phi_9(1)+16*pi/lamda_1*sin(theta)*d;

phi_10_1 = phi_10(1)+18*pi/lamda_1*sin(theta)*d;

phi_11_1 = phi_11(1)+20*pi/lamda_1*sin(theta)*d;

phi_12_1 = phi_12(1)+22*pi/lamda_1*sin(theta)*d;

phi_13_1 = phi_13(1)+24*pi/lamda_1*sin(theta)*d;

phi_14_1 = phi_14(1)+26*pi/lamda_1*sin(theta)*d;

phi_15_1 = phi_15(1)+28*pi/lamda_1*sin(theta)*d;

phi_16_1 = phi_16(1)+30*pi/lamda_1*sin(theta)*d;


phi_1_2 = phi_1(2)+0*pi/lamda_2*sin(theta)*d;

phi_2_2 = phi_2(2)+2*pi/lamda_2*sin(theta)*d;

phi_3_2 = phi_3(2)+4*pi/lamda_2*sin(theta)*d;

phi_4_2 = phi_4(2)+6*pi/lamda_2*sin(theta)*d;

phi_5_2 = phi_5(2)+8*pi/lamda_2*sin(theta)*d;

phi_6_2 = phi_6(2)+10*pi/lamda_2*sin(theta)*d;

phi_7_2 = phi_7(2)+12*pi/lamda_2*sin(theta)*d;

phi_8_2 = phi_8(2)+14*pi/lamda_2*sin(theta)*d;

phi_9_2 = phi_9(2)+16*pi/lamda_2*sin(theta)*d;

phi_10_2 = phi_10(2)+18*pi/lamda_2*sin(theta)*d;

phi_11_2 = phi_11(2)+20*pi/lamda_2*sin(theta)*d;

phi_12_2 = phi_12(2)+22*pi/lamda_2*sin(theta)*d;

phi_13_2 = phi_13(2)+24*pi/lamda_2*sin(theta)*d;
```

```
phi_14_2 = phi_14(2)+26*pi/lamda_2*sin(theta)*d;
phi_15_2 = phi_15(2)+28*pi/lamda_2*sin(theta)*d;
phi_16_2 = phi_16(2)+30*pi/lamda_2*sin(theta)*d;

new = chebwin(16,18);
A1 =
(abs(exp(1i*phi_1_2)+exp(1i*phi_2_2)+exp(1i*phi_3_2)+exp(1i*phi_4_2)+exp(1i*phi_5_2)+exp(1i*phi_6_2)+exp(1i*phi_7_2)+exp(1i*phi_8_2)+exp(1i*phi_9_2)+exp(1i*phi_10_2)+exp(1i*phi_11_2)+exp(1i*phi_12_2)+exp(1i*phi_13_2)+exp(1i*phi_14_2)+exp(1i*phi_15_2)+exp(1i*phi_16_2))).^2;
A2 =
(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(new(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(14)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)))).^2;
new = chebwin(16,22);
A3 =
(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(new(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(14)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)))).^2;
```

```
new = chebwin(16,30);
A4 =
(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)
*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi
_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(n
ew(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(1
4)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)))).^2;


figure(2)
polarplot(theta, A1);
hold on
polarplot(theta, A2);
polarplot(theta, A3);
polarplot(theta, A4);
title("16 Element Array Factor Polar Pattern")
legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")

figure(4)
plot(theta/pi*180, 10*log10(A1), 'LineWidth',1.7);
hold on
plot(theta/pi*180, 10*log10(A2), 'LineWidth',1.7);
plot(theta/pi*180, 10*log10(A3), 'LineWidth',1.7);
```

```
    plot(theta/pi*180, 10*log10(A4), 'LineWidth',1.7);
    title("16 Element Array Factor Pattern")
    ylabel("Array Factor (dB)")
    xlabel("Angle (°)")
    legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")

else if (peaks_check == 16)
    theta = linspace(-pi/2,pi/2,10001);
    c = 3e8;
    f1 = f_peak_1(1);
    f2 = f_peak_1(2);
    lamda_1 = c/f1;
    lamda_2 = c/f2;
    lamda = c/(150e9);
    d = lamda/2;

    phi_1_1 = phi_1(1)+0*pi/lamda_1*sin(theta)*d;
    phi_2_1 = phi_2(1)+2*pi/lamda_1*sin(theta)*d;
    phi_3_1 = phi_3(1)+4*pi/lamda_1*sin(theta)*d;
    phi_4_1 = phi_4(1)+6*pi/lamda_1*sin(theta)*d;
    phi_5_1 = phi_5(1)+8*pi/lamda_1*sin(theta)*d;
    phi_6_1 = phi_6(1)+10*pi/lamda_1*sin(theta)*d;
    phi_7_1 = phi_7(1)+12*pi/lamda_1*sin(theta)*d;
```

```
phi_8_1 = phi_8(1)+14*pi/lamda_1*sin(theta)*d;

phi_1_2 = phi_1(2)+0*pi/lamda_2*sin(theta)*d;
phi_2_2 = phi_2(2)+2*pi/lamda_2*sin(theta)*d;
phi_3_2 = phi_3(2)+4*pi/lamda_2*sin(theta)*d;
phi_4_2 = phi_4(2)+6*pi/lamda_2*sin(theta)*d;
phi_5_2 = phi_5(2)+8*pi/lamda_2*sin(theta)*d;
phi_6_2 = phi_6(2)+10*pi/lamda_2*sin(theta)*d;
phi_7_2 = phi_7(2)+12*pi/lamda_2*sin(theta)*d;
phi_8_2 = phi_8(2)+14*pi/lamda_2*sin(theta)*d;

phi_8_all = [phi_1(2),phi_2(2),phi_3(2),phi_4(2),phi_5(2),phi_6(2),phi_7(2),phi_8(2)]

new1 = chebwin(8,18);
A1 =
(abs(exp(1i*phi_1_2)+exp(1i*phi_2_2)+exp(1i*phi_3_2)+exp(1i*phi_4_2)+exp(1i*phi_5_2)+e
xp(1i*phi_6_2)+exp(1i*phi_7_2)+exp(1i*phi_8_2))).^2;
A2 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2)+new1(5)*exp(1i*phi_5_2)+new1(6)*exp(1i*phi_6_2)+new1(7)*exp(1i*phi_
7_2)+new1(8)*exp(1i*phi_8_2))).^2;
new1 = chebwin(8,22);
```

```
A3 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2)+new1(5)*exp(1i*phi_5_2)+new1(6)*exp(1i*phi_6_2)+new1(7)*exp(1i*phi_
7_2)+new1(8)*exp(1i*phi_8_2))).^2;
new1 = chebwin(8,30);
A4 =
(abs(new1(1)*exp(1i*phi_1_2)+new1(2)*exp(1i*phi_2_2)+new1(3)*exp(1i*phi_3_2)+new1(4)
*exp(1i*phi_4_2)+new1(5)*exp(1i*phi_5_2)+new1(6)*exp(1i*phi_6_2)+new1(7)*exp(1i*phi_
7_2)+new1(8)*exp(1i*phi_8_2))).^2;

figure(67)
polarplot(theta, A1);
hold on
polarplot(theta, A2);
polarplot(theta, A3);
polarplot(theta, A4);
title("8 Element Array Factor Polar Pattern")
legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")
figure(3)
plot(theta/pi*180, 10*log10(A1), 'LineWidth',1.7);
hold on
plot(theta/pi*180, 10*log10(A2), 'LineWidth',1.7);
plot(theta/pi*180, 10*log10(A3), 'LineWidth',1.7);
```

```
    plot(theta/pi*180, 10*log10(A4), 'LineWidth',1.7);
    title("8 Element Array Factor Pattern")
    ylabel("Array Factor (dB)")
    xlabel("Angle (°)")
    legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")

else
    theta = linspace(-pi/2,pi/2,10001);
    c = 3e8;
    f1 = f_peak_1(1);
    f2 = f_peak_1(2);
    lamda_1 = c/f1;
    lamda_2 = c/f2;
    lamda = c/(150e9);
    d = lamda/2;

    phi_1_1 = phi_1(1)+0*pi/lamda_1*sin(theta)*d;
    phi_2_1 = phi_2(1)+2*pi/lamda_1*sin(theta)*d;
    phi_3_1 = phi_3(1)+4*pi/lamda_1*sin(theta)*d;
    phi_4_1 = phi_4(1)+6*pi/lamda_1*sin(theta)*d;
    phi_5_1 = phi_5(1)+8*pi/lamda_1*sin(theta)*d;
    phi_6_1 = phi_6(1)+10*pi/lamda_1*sin(theta)*d;
    phi_7_1 = phi_7(1)+12*pi/lamda_1*sin(theta)*d;
```

```
phi_8_1 = phi_8(1)+14*pi/lamda_1*sin(theta)*d;

phi_9_1 = phi_9(1)+16*pi/lamda_1*sin(theta)*d;

phi_10_1 = phi_10(1)+18*pi/lamda_1*sin(theta)*d;

phi_11_1 = phi_11(1)+20*pi/lamda_1*sin(theta)*d;

phi_12_1 = phi_12(1)+22*pi/lamda_1*sin(theta)*d;

phi_13_1 = phi_13(1)+24*pi/lamda_1*sin(theta)*d;

phi_14_1 = phi_14(1)+26*pi/lamda_1*sin(theta)*d;

phi_15_1 = phi_15(1)+28*pi/lamda_1*sin(theta)*d;

phi_16_1 = phi_16(1)+30*pi/lamda_1*sin(theta)*d;

phi_17_1 = phi_17(1)+32*pi/lamda_1*sin(theta)*d;

phi_18_1 = phi_18(1)+34*pi/lamda_1*sin(theta)*d;

phi_19_1 = phi_19(1)+36*pi/lamda_1*sin(theta)*d;

phi_20_1 = phi_20(1)+38*pi/lamda_1*sin(theta)*d;

phi_21_1 = phi_21(1)+40*pi/lamda_1*sin(theta)*d;

phi_22_1 = phi_22(1)+42*pi/lamda_1*sin(theta)*d;

phi_23_1 = phi_23(1)+44*pi/lamda_1*sin(theta)*d;

phi_24_1 = phi_24(1)+46*pi/lamda_1*sin(theta)*d;

phi_25_1 = phi_25(1)+48*pi/lamda_1*sin(theta)*d;

phi_26_1 = phi_26(1)+50*pi/lamda_1*sin(theta)*d;

phi_27_1 = phi_27(1)+52*pi/lamda_1*sin(theta)*d;

phi_28_1 = phi_28(1)+54*pi/lamda_1*sin(theta)*d;

phi_29_1 = phi_29(1)+56*pi/lamda_1*sin(theta)*d;

phi_30_1 = phi_30(1)+58*pi/lamda_1*sin(theta)*d;
```

```
phi_31_1 = phi_31(1)+60*pi/lamda_1*sin(theta)*d;
phi_32_1 = phi_32(1)+62*pi/lamda_1*sin(theta)*d;

phi_1_2 = phi_1(2)+0*pi/lamda_2*sin(theta)*d;
phi_2_2 = phi_2(2)+2*pi/lamda_2*sin(theta)*d;
phi_3_2 = phi_3(2)+4*pi/lamda_2*sin(theta)*d;
phi_4_2 = phi_4(2)+6*pi/lamda_2*sin(theta)*d;
phi_5_2 = phi_5(2)+8*pi/lamda_2*sin(theta)*d;
phi_6_2 = phi_6(2)+10*pi/lamda_2*sin(theta)*d;
phi_7_2 = phi_7(2)+12*pi/lamda_2*sin(theta)*d;
phi_8_2 = phi_8(2)+14*pi/lamda_2*sin(theta)*d;
phi_9_2 = phi_9(2)+16*pi/lamda_2*sin(theta)*d;
phi_10_2 = phi_10(2)+18*pi/lamda_2*sin(theta)*d;
phi_11_2 = phi_11(2)+20*pi/lamda_2*sin(theta)*d;
phi_12_2 = phi_12(2)+22*pi/lamda_2*sin(theta)*d;
phi_13_2 = phi_13(2)+24*pi/lamda_2*sin(theta)*d;
phi_14_2 = phi_14(2)+26*pi/lamda_2*sin(theta)*d;
phi_15_2 = phi_15(2)+28*pi/lamda_2*sin(theta)*d;
phi_16_2 = phi_16(2)+30*pi/lamda_2*sin(theta)*d;
phi_17_2 = phi_17(2)+32*pi/lamda_2*sin(theta)*d;
phi_18_2 = phi_18(2)+34*pi/lamda_2*sin(theta)*d;
phi_19_2 = phi_19(2)+36*pi/lamda_2*sin(theta)*d;
phi_20_2 = phi_20(2)+38*pi/lamda_2*sin(theta)*d;
```

```
phi_21_2 = phi_21(2)+40*pi/lamda_2*sin(theta)*d;

phi_22_2 = phi_22(2)+42*pi/lamda_2*sin(theta)*d;

phi_23_2 = phi_23(2)+44*pi/lamda_2*sin(theta)*d;

phi_24_2 = phi_24(2)+46*pi/lamda_2*sin(theta)*d;

phi_25_2 = phi_25(2)+48*pi/lamda_2*sin(theta)*d;

phi_26_2 = phi_26(2)+50*pi/lamda_2*sin(theta)*d;

phi_27_2 = phi_27(2)+52*pi/lamda_2*sin(theta)*d;

phi_28_2 = phi_28(2)+54*pi/lamda_2*sin(theta)*d;

phi_29_2 = phi_29(2)+56*pi/lamda_2*sin(theta)*d;

phi_30_2 = phi_30(2)+58*pi/lamda_2*sin(theta)*d;

phi_31_2 = phi_31(2)+60*pi/lamda_2*sin(theta)*d;

phi_32_2 = phi_32(2)+62*pi/lamda_2*sin(theta)*d;


new = chebwin(32,18);

A1 =

(abs(exp(1i*phi_1_2)+exp(1i*phi_2_2)+exp(1i*phi_3_2)+exp(1i*phi_4_2)+exp(1i*phi_5_2)+e
xp(1i*phi_6_2)+exp(1i*phi_7_2)+exp(1i*phi_8_2)+exp(1i*phi_9_2)+exp(1i*phi_10_2)+exp(1i
*phi_11_2)+exp(1i*phi_12_2)+exp(1i*phi_13_2)+exp(1i*phi_14_2)+exp(1i*phi_15_2)+exp(1i
*phi_16_2)+exp(1i*phi_17_2)+exp(1i*phi_18_2)+exp(1i*phi_19_2)+exp(1i*phi_20_2)+exp(1i
*phi_21_2)+exp(1i*phi_22_2)+exp(1i*phi_23_2)+exp(1i*phi_24_2)+exp(1i*phi_25_2)+exp(1i
*phi_26_2)+exp(1i*phi_27_2)+exp(1i*phi_28_2)+exp(1i*phi_29_2)+exp(1i*phi_30_2)+exp(1i
*phi_31_2)+exp(1i*phi_32_2))).^2;
```

%A2 =

(abs(exp(1i*phi_1_1)+exp(1i*phi_2_1)+exp(1i*phi_3_1)+exp(1i*phi_4_1)+exp(1i*phi_5_1)+exp(1i*phi_6_1)+exp(1i*phi_7_1)+exp(1i*phi_8_1)+exp(1i*phi_9_1)+exp(1i*phi_10_1)+exp(1i*phi_11_1)+exp(1i*phi_12_1)+exp(1i*phi_13_1)+exp(1i*phi_14_1)+exp(1i*phi_15_1)+exp(1i*phi_16_1)+exp(1i*phi_17_1)+exp(1i*phi_18_1)+exp(1i*phi_19_1)+exp(1i*phi_20_1)+exp(1i*phi_21_1)+exp(1i*phi_22_1)+exp(1i*phi_23_1)+exp(1i*phi_24_1)+exp(1i*phi_25_1)+exp(1i*phi_26_1)+exp(1i*phi_27_1)+exp(1i*phi_28_1)+exp(1i*phi_29_1)+exp(1i*phi_30_1)+exp(1i*phi_31_1)+exp(1i*phi_32_1))).^2;

%A2 =

(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(new(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(14)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)+(new(17)*exp(1i*phi_17_2))+(new(18)*exp(1i*phi_18_2))+(new(19)*exp(1i*phi_19_2))+(new(20)*exp(1i*phi_20_2))+(new(21)*exp(1i*phi_21_2))+(new(22)*exp(1i*phi_22_2))+(new(23)*exp(1i*phi_23_2))+(new(24)*exp(1i*phi_24_2))+(new(25)*exp(1i*phi_25_2))+(new(26)*exp(1i*phi_26_2))+(new(27)*exp(1i*phi_27_2))+(new(28)*exp(1i*phi_28_2))+(new(29)*exp(1i*phi_29_2))+(new(30)*exp(1i*phi_30_2))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_32_2))))).^2;

new = chebwin(32,22);

%A3 =

(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)

*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(new(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(14)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)+(new(17)*exp(1i*phi_17_2))+(new(18)*exp(1i*phi_18_2))+(new(19)*exp(1i*phi_19_2))+(new(20)*exp(1i*phi_20_2))+(new(21)*exp(1i*phi_21_2))+(new(22)*exp(1i*phi_22_2))+(new(23)*exp(1i*phi_23_2))+(new(24)*exp(1i*phi_24_2))+(new(25)*exp(1i*phi_25_2))+(new(26)*exp(1i*phi_26_2))+(new(27)*exp(1i*phi_27_2))+(new(28)*exp(1i*phi_28_2))+(new(29)*exp(1i*phi_29_2))+(new(30)*exp(1i*phi_30_2))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_32_2))))).^2;

new = chebwin(32,30);

%A4 =

(abs((new(1)*exp(1i*phi_1_2))+(new(2)*exp(1i*phi_2_2))+(new(3)*exp(1i*phi_3_2))+(new(4)*exp(1i*phi_4_2))+(new(5)*exp(1i*phi_5_2))+(new(6)*exp(1i*phi_6_2))+(new(7)*exp(1i*phi_7_2))+(new(8)*exp(1i*phi_8_2))+(new(9)*exp(1i*phi_9_2))+(new(10)*exp(1i*phi_10_2))+(new(11)*exp(1i*phi_11_2))+(new(12)*exp(1i*phi_12_2))+(new(13)*exp(1i*phi_13_2))+(new(14)*exp(1i*phi_14_2))+(new(15)*exp(1i*phi_15_2))+(new(16)*exp(1i*phi_16_2)+(new(17)*exp(1i*phi_17_2))+(new(18)*exp(1i*phi_18_2))+(new(19)*exp(1i*phi_19_2))+(new(20)*exp(1i*phi_20_2))+(new(21)*exp(1i*phi_21_2))+(new(22)*exp(1i*phi_22_2))+(new(23)*exp(1i*phi_23_2))+(new(24)*exp(1i*phi_24_2))+(new(25)*exp(1i*phi_25_2))+(new(26)*exp(1i*phi_26_2))+(new(27)*exp(1i*phi_27_2))+(new(28)*exp(1i*phi_28_2))+(new(29)*exp(1i*phi_29_2))+(new(30)*exp(1i*phi_30_2))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_32_2))))).^2;

```
phi_32_all =
[phi_1(2),phi_2(2),phi_3(2),phi_4(2),phi_5(2),phi_6(2),phi_7(2),phi_8(2),phi_9(2),phi_10(2),ph
i_11(2),phi_12(2),phi_13(2),phi_14(2),phi_15(2),phi_16(2),phi_17(2),phi_18(2),phi_19(2),phi_
20(2),phi_21(2),phi_22(2),phi_23(2),phi_24(2),phi_25(2),phi_26(2),phi_27(2),phi_28(2),phi_29
(2),phi_30(2),phi_31(2),phi_32(2)]
[M,I] = max(A1)
theta(I)*180/pi


%       figure(87)
%       polarplot(theta, A1);
%       hold on
%       polarplot(theta, A2);
%       polarplot(theta, A3);
%       polarplot(theta, A4);
%       title("32 Element Array Factor Pattern")
%       legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")
%       hold off
figure(89)
plot(theta/pi*180, 10*log10(A1), 'LineWidth',1.7);
hold on
%plot(theta/pi*180, 10*log10(A2), 'LineWidth',1.7);
%plot(theta/pi*180, 10*log10(A3), 'LineWidth',1.7);
%plot(theta/pi*180, 10*log10(A4), 'LineWidth',1.7);
```

```
        title("32 Element Array Factor Pattern")
        ylabel("Array Factor (dB)")
        xlabel("Angle (°)")
        legend("0dB Suppression","18dB Suppression","22dB Suppresion","30dB Suppression")
        hold off
    end
    end
end
```

line_fft.m

```
function [spectrum, frequency, phase, f_peak, num_of_peaks] =
line_fft(input_data,data_length,fs)
phase = zeros(1,2);
f_peak = zeros(1,2);
Y = fft(input_data);
P2 = abs(Y/data_length);
spectrum = P2(1:(data_length/2+1));
spectrum(2:end-1) = 1*spectrum(2:end-1);
spectrum = (10*log(spectrum));
frequency = fs*(0:(data_length/2))/data_length;
num_of_peaks = 0;
peak_found = 0;
for i = 1:round(length(frequency)*(4e11)/fs)
   peak_amplitude = 0;
   if spectrum(i)+10> 0
      if peak_found ~= 1
         peak_found = 1;
         num_of_peaks = num_of_peaks + 1;
         peak_amplitude = spectrum(i);
         f_peak(num_of_peaks) = frequency(i);
         phase(num_of_peaks) = angle(Y(i));
```

```
        else if spectrum(i) > peak_amplitude
                peak_amplitude = spectrum(i);
                f_peak(num_of_peaks) = frequency(i);
                phase(num_of_peaks) = angle(Y(i));
            end
        end
    else
        peak_found = 0;
    end
end


end
```

Chebyshev_and_Array_Factor.m

```
%%Constants

theta = linspace(-pi/2,pi/2,10001);

F = linspace(25,35,21);

%% 4 Element


phi_4_25 = [2.719627298858368,-

1.447888837063152,0.827840372740135,2.934271670673686];

lamb_4_25 = 0.002017406012658;

d_4_25 = 1.000000000000000e-03;

[AF_4_25] = Array_Fact(phi_4_25,lamb_4_25,d_4_25);


phi_4_25_5 = [3.132718777090693,-

1.353262049291329,0.730818027678871,2.656355861230414];

lamb_4_25_5 = 0.001992188437500;

d_4_25_5 = 1.000000000000000e-03;

[AF_4_25_5] = Array_Fact(phi_4_25_5,lamb_4_25_5,d_4_25_5);


phi_4_26 = [-2.785267576311752,-

1.246138931285403,0.621404606022291,2.397223593357264];

lamb_4_26 = 0.001970634621329;

d_4_26 = 1.000000000000000e-03;

[AF_4_26] = Array_Fact(phi_4_26,lamb_4_26,d_4_26);
```

```
phi_4_26_5 = [-2.476437591304110,-
1.121090189044362,0.508597618262166,2.114955576161288];
lamb_4_26_5 = 0.001970634621329;
d_4_26_5 = 1.000000000000000e-03;
[AF_4_26_5] = Array_Fact(phi_4_26_5,lamb_4_26_5,d_4_26_5);

phi_4_27 = [-2.194851265744545,-
0.975089174954274,0.403702169776188,1.791147305914651];
lamb_4_27 = 0.001982893623639;
d_4_27 = 1.000000000000000e-03;
[AF_4_27] = Array_Fact(phi_4_27,lamb_4_27,d_4_27);

phi_4_27_5 = [-1.881889069297370,-
0.811351117966902,0.313409679373645,1.444775751217042];
lamb_4_27_5 = 0.002004717924528;
d_4_27_5 = 1.000000000000000e-03;
[AF_4_27_5] = Array_Fact(phi_4_27_5,lamb_4_27_5,d_4_27_5);

phi_4_28 = [-1.520126916391590,-
0.640480830067336,0.239164808749194,1.118811107764079];
lamb_4_28 = 0.002027027980922;
d_4_28 = 1.000000000000000e-03;
```

[AF_4_28] = Array_Fact(phi_4_28,lamb_4_28,d_4_28);

phi_4_28_5 = [1.941482079426126,2.593746411342773,-3.029768191674550,-2.361426124565150];

lamb_4_28_5 = 0.001992188437500;

d_4_28_5 = 1.000000000000000e-03;

[AF_4_28_5] = Array_Fact(phi_4_28_5,lamb_4_28_5,d_4_28_5);

phi_4_29 = [2.298968013868418,2.706791208910955,3.124435879094163,-2.713262831190881];

lamb_4_29 = 0.001964561787365;

d_4_29 = 1.000000000000000e-03;

[AF_4_29] = Array_Fact(phi_4_29,lamb_4_29,d_4_29);

phi_4_29_5 = [2.698360196373758,2.838190636745065,3.001377638229532,-3.044162221770459];

lamb_4_29_5 = 0.001946565801527;

d_4_29_5 = 1.000000000000000e-03;

[AF_4_29_5] = Array_Fact(phi_4_29_5,lamb_4_29_5,d_4_29_5);

phi_4_30

=[3.102122464587827,2.985323541108383,2.889158227860431,2.948790211355917];

lamb_4_30 = 0.001931819090909;

```
d_4_30 = 1.000000000000000e-03;

[AF_4_30] = Array_Fact(phi_4_30,lamb_4_30,d_4_30);


phi_4_30_5 =[-
2.826118202377861,3.134409669597199,2.803735978498883,2.681743071574279];

lamb_4_30_5 = 0.001928896520424;

d_4_30_5 = 1.000000000000000e-03;

[AF_4_30_5] = Array_Fact(phi_4_30_5,lamb_4_30_5,d_4_30_5);


phi_4_31 =[-2.518066533037727,-
3.008065476681234,2.735057029150387,2.399751628372993];

lamb_4_31 = 0.001934750531108;

d_4_31 = 1.000000000000000e-03;

[AF_4_31] = Array_Fact(phi_4_31,lamb_4_31,d_4_31);


phi_4_31_5 =[-2.225300879935058,-
2.884544965056753,2.681020332807425,2.082082109213580];

lamb_4_31_5 = 0.001952527718224;

d_4_31_5 = 1.000000000000000e-03;

[AF_4_31_5] = Array_Fact(phi_4_31_5,lamb_4_31_5,d_4_31_5);


phi_4_32 = [-1.898929908901789,-
2.778575171641220,2.624963375295075,1.745318495679149];
```

```
lamb_4_32 = 0.001973685139319;

d_4_32 = 1.000000000000000e-03;

[AF_4_32] = Array_Fact(phi_4_32,lamb_4_32,d_4_32);


phi_4_32_5 = [1.563844714282609,0.516180858370870,-0.646410502789524,-
1.705653618567265];

lamb_4_32_5 = 0.001940640182648;

d_4_32_5 = 1.000000000000000e-03;

[AF_4_32_5] = Array_Fact(phi_4_32_5,lamb_4_32_5,d_4_32_5);


phi_4_33 =[1.883537862106561,0.674475362754779,-0.759147117995294,-
2.046863703962902];

lamb_4_33 = 0.001914415315315;

d_4_33 = 1.000000000000000e-03;

[AF_4_33] = Array_Fact(phi_4_33,lamb_4_33,d_4_33);


phi_4_33_5 = [2.238836897648335,0.819518316806778,-0.853322305295026,-
2.402297726027775];

lamb_4_33_5 = 0.001897322321429;

d_4_33_5 = 1.000000000000000e-03;

[AF_4_33_5] = Array_Fact(phi_4_33_5,lamb_4_33_5,d_4_33_5);
```

```
phi_4_34 = [2.630913320356338,0.948103539760524,-0.936263836945824,-
2.734937874542887];
lamb_4_34 = 0.001888889777778;
d_4_34 = 1.000000000000000e-03;
[AF_4_34] = Array_Fact(phi_4_34,lamb_4_34,d_4_34);

phi_4_34_5 = [3.019884998832454,1.058035655513161,-1.013865102372784,-
3.032292504023382];
lamb_4_34_5 = 0.001888889777778;
d_4_34_5 = 1.000000000000000e-03;
[AF_4_34_5] = Array_Fact(phi_4_34_5,lamb_4_34_5,d_4_34_5);

phi_4_35 = [-2.912323624290464,1.156840415594972,-
1.094411908693769,2.972505070354418];
lamb_4_35 = 0.001894503120357;
d_4_35 = 1.000000000000000e-03;
[AF_4_35] = Array_Fact(phi_4_35,lamb_4_35,d_4_35);

[Peak_4_25,I_4_25] = max(AF_4_25);
Angle_4_25=theta(I_4_25);
[Peak_4_25_5,I_4_25_5] = max(AF_4_25_5);
Angle_4_25_5 = theta(I_4_25_5);
[Peak_4_26,I_4_26] = max(AF_4_26);
```

```
Angle_4_26=theta(I_4_26);
[Peak_4_26_5,I_4_26_5] = max(AF_4_26_5);
Angle_4_26_5 = theta(I_4_26_5);
[Peak_4_27,I_4_27] = max(AF_4_27);
Angle_4_27=theta(I_4_27);
[Peak_4_27_5,I_4_27_5] = max(AF_4_27_5);
Angle_4_27_5=theta(I_4_27_5);
[Peak_4_28,I_4_28] = max(AF_4_28);
Angle_4_28=theta(I_4_28);
[Peak_4_28_5,I_4_28_5] = max(AF_4_28_5);
Angle_4_28_5=theta(I_4_28_5);
[Peak_4_29,I_4_29] = max(AF_4_29);
Angle_4_29=theta(I_4_29);
[Peak_4_29_5,I_4_29_5] = max(AF_4_29_5);
Angle_4_29_5=theta(I_4_29_5);
[Peak_4_30,I_4_30] = max(AF_4_30);
Angle_4_30=theta(I_4_30);
[Peak_4_30_5,I_4_30_5] = max(AF_4_30_5);
Angle_4_30_5=theta(I_4_30_5);
[Peak_4_31,I_4_31] = max(AF_4_31);
Angle_4_31=theta(I_4_31);
[Peak_4_31_5,I_4_31_5] = max(AF_4_31_5);
Angle_4_31_5=theta(I_4_31_5);
```

```
[Peak_4_32,I_4_32] = max(AF_4_32);

Angle_4_32=theta(I_4_32);

[Peak_4_32_5,I_4_32_5] = max(AF_4_32_5);

Angle_4_32_5=theta(I_4_32_5);

[Peak_4_33,I_4_33] = max(AF_4_33);

Angle_4_33 = theta(I_4_33);

[Peak_4_33_5,I_4_33_5] = max(AF_4_33_5);

Angle_4_33_5=theta(I_4_33_5);

[Peak_4_34,I_4_34] = max(AF_4_34);

Angle_4_34=theta(I_4_34);

[Peak_4_34_5,I_4_34_5] = max(AF_4_34_5);

Angle_4_34_5=theta(I_4_34_5);

[Peak_4_35,I_4_35] = max(AF_4_35);

Angle_4_35=theta(I_4_35);


Angles_4 = [Angle_4_25,

Angle_4_25_5,Angle_4_26,Angle_4_26_5,Angle_4_27,Angle_4_27_5,Angle_4_28,Angle_4_2

8_5,Angle_4_29,Angle_4_29_5,Angle_4_30,Angle_4_30_5,Angle_4_31,Angle_4_31_5,Angle_

4_32,Angle_4_32_5,Angle_4_33,Angle_4_33_5,Angle_4_34,Angle_4_34_5,Angle_4_35];


%2D plot

figure(2)

plot(theta/pi*180,10*log10(AF_4_25))
```

```
hold on
plot(theta/pi*180,10*log10(AF_4_25_5))
plot(theta/pi*180,10*log10(AF_4_26))
plot(theta/pi*180,10*log10(AF_4_26_5))
plot(theta/pi*180,10*log10(AF_4_27))
plot(theta/pi*180,10*log10(AF_4_27_5))
plot(theta/pi*180,10*log10(AF_4_28))
plot(theta/pi*180,10*log10(AF_4_28_5))
plot(theta/pi*180,10*log10(AF_4_29))
plot(theta/pi*180,10*log10(AF_4_29_5))
plot(theta/pi*180,10*log10(AF_4_30))
plot(theta/pi*180,10*log10(AF_4_30_5))
plot(theta/pi*180,10*log10(AF_4_31))
plot(theta/pi*180,10*log10(AF_4_31_5))
plot(theta/pi*180,10*log10(AF_4_32))
plot(theta/pi*180,10*log10(AF_4_32_5))
plot(theta/pi*180,10*log10(AF_4_33))
plot(theta/pi*180,10*log10(AF_4_33_5))
plot(theta/pi*180,10*log10(AF_4_34))
plot(theta/pi*180,10*log10(AF_4_34_5))
plot(theta/pi*180,10*log10(AF_4_35))
title("4 Element Array Factor: Shift from 25GHz to 35GHz")
ylabel("Array Factor (dB)")
```

```
xlabel("Angle (°)")
hold off


figure(56)
polarplot(theta,AF_4_25)
hold on
polarplot(theta,AF_4_25_5)
polarplot(theta,AF_4_26)
polarplot(theta,AF_4_26_5)
polarplot(theta,AF_4_27)
polarplot(theta,AF_4_27_5)
polarplot(theta,AF_4_28)
polarplot(theta,AF_4_28_5)
polarplot(theta,AF_4_29)
polarplot(theta,AF_4_29_5)
polarplot(theta,AF_4_30)
polarplot(theta,AF_4_30_5)
polarplot(theta,AF_4_31)
polarplot(theta,AF_4_31_5)
polarplot(theta,AF_4_32)
polarplot(theta,AF_4_32_5)
polarplot(theta,AF_4_33)
polarplot(theta,AF_4_33_5)
```

```
polarplot(theta,AF_4_34)
polarplot(theta,AF_4_34_5)
polarplot(theta,AF_4_35)
title("4 Element Array Factor: Shift from 25GHz to 35GHz")
hold off


%3D
AF_4 = [10*log10(AF_4_25);
10*log10(AF_4_25_5);10*log10(AF_4_26);10*log10(AF_4_26_5);10*log10(AF_4_27);10*log
10(AF_4_27_5);10*log10(AF_4_28);
10*log10(AF_4_28_5);10*log10(AF_4_29);10*log10(AF_4_29_5);10*log10(AF_4_30);10*log
10(AF_4_30_5);10*log10(AF_4_31);
10*log10(AF_4_31_5);10*log10(AF_4_32);10*log10(AF_4_32_5);10*log10(AF_4_33);10*log
10(AF_4_33_5);10*log10(AF_4_34);10*log10(AF_4_34_5);10*log10(AF_4_35)];

%AF_4 = [AF_4_25; AF_4_25_5;AF_4_26;AF_4_26_5;AF_4_27;AF_4_27_5;AF_4_28;
AF_4_28_5;AF_4_29;AF_4_29_5;AF_4_30;AF_4_30_5;AF_4_31;
AF_4_31_5;AF_4_32;AF_4_32_5;AF_4_33;AF_4_33_5;AF_4_34;AF_4_34_5;AF_4_35];

figure(12)
surf(theta/pi*180,F,AF_4)
hold on
```

```
title('4 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off
figure(70)
mesh(theta/pi*180,F,AF_4)
hold on
title('4 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off


%% 8 Element
phi_8_25 = [0.173112296175573,2.34615816341953,-
1.71554015216587,0.471173729074717,2.68588981268957,-
1.41773794876660,0.799455384319947,2.97687882275340];
lamb_8_25 = 0.002056452580645;
d_8_25 = 1.000000000000000e-03;
[AF_8_25] = Array_Fact(phi_8_25,lamb_8_25,d_8_25);
```

```
phi_8_25_5 =[0.540404303358071,2.46327525570085,-
1.80805628438369,0.171388946365696,2.13489270381454,-2.13337066901039,-
0.212147346727653,1.81777591264909];
lamb_8_25_5 = 0.002043270192308;
d_8_25_5 = 1.000000000000000e-03;
[AF_8_25_5] = Array_Fact(phi_8_25_5,lamb_8_25_5,d_8_25_5);

phi_8_26 = [0.887863718330820,2.58932127976843,-1.90247044001532,-
0.128509163448800,1.59392550928737,-2.89188252718297,-
1.15362129647822,0.585855383127976];
lamb_8_26 = 0.002036742172524;
d_8_26 = 1.000000000000000e-03;
[AF_8_26] = Array_Fact(phi_8_26,lamb_8_26,d_8_26);

phi_8_26_5 = [1.22338637331251,2.71664217762238,-1.99843044899235,-
0.435096063033003,1.08030391116999,2.60035211825221,-2.10352557208130,-
0.582599270305962];
lamb_8_26_5 = 0.002030255732484;
d_8_26_5 = 1.000000000000000e-03;
[AF_8_26_5] = Array_Fact(phi_8_26_5,lamb_8_26_5,d_8_26_5);
```

```
phi_8_27 = [1.55371784119732,2.85074505529313,-2.09345287798921,-
0.753666147985266,0.567881383467574,1.85572414843154,-3.11344455293504,-
1.76916859341871];
lamb_8_27 = 0.002027027980922;
d_8_27 = 1.000000000000000e-03;
[AF_8_27] = Array_Fact(phi_8_27,lamb_8_27,d_8_27);

phi_8_27_5 = [1.89536440790651,2.99081473625460,-2.18517109289252,-
1.07759416619259,0.0292883034149129,1.12534098910879,2.21170369764490,-
2.97603185937538];
lamb_8_27_5 = 0.002027027980922;
d_8_27_5 = 1.000000000000000e-03;
[AF_8_27_5] = Array_Fact(phi_8_27_5,lamb_8_27_5,d_8_27_5);

phi_8_28 = [2.24978431710837,3.12942994096968,-2.27410940072570,-1.39446253220688,-
0.514816004882035,0.364829773283484,1.24447568097769,2.12412202065147];
lamb_8_28 = 0.002027027980922;
d_8_28 = 1.000000000000000e-03;
[AF_8_28] = Array_Fact(phi_8_28,lamb_8_28,d_8_28);

phi_8_28_5 = [-
0.552825352324880,0.106885901466344,0.768506940725844,1.42972916503317,2.089099714
01639,2.74783379157489,-2.87536690315816,-2.21413184668271];
```

```
lamb_8_28_5 = 0.002014218957346;

d_8_28_5 = 1.000000000000000e-03;

[AF_8_28_5] = Array_Fact(phi_8_28_5,lamb_8_28_5,d_8_28_5);


phi_8_29 = [-
0.204204864217001,0.228965834333231,0.667383064223498,1.10923994762372,1.548996527
88566,1.98568924857819,2.42509002834010,2.87248026981661];

lamb_8_29 = 0.002001570800628;

d_8_29 = 1.000000000000000e-03;

[AF_8_29] = Array_Fact(phi_8_29,lamb_8_29,d_8_29);


phi_8_29_5 =
[0.155632332231460,0.356798043128517,0.568768934412447,0.795996901269239,1.0299939
8558239,1.26128674255768,1.48117124919194,1.69784111179126];

lamb_8_29_5 =0.001992188437500;

d_8_29_5 = 1.000000000000000e-03;

[AF_8_29_5] = Array_Fact(phi_8_29_5,lamb_8_29_5,d_8_29_5);


phi_8_30 =
[0.515867161718880,0.486234670830205,0.470036703942497,0.489630347004547,0.5173313
66111394,0.515867188967302,0.486234643384915,0.470036731744352];

lamb_8_30 =0.001982893623639;

d_8_30 =1.000000000000000e-03;
```

[AF_8_30] = Array_Fact(phi_8_30,lamb_8_30,d_8_30);

phi_8_30_5 =

[0.862887878800164,0.617171858547003,0.375652312531829,0.186691411914037,-0.0167157802789208,-0.273748542356300,-0.508068092964602,-0.695469934055709];

lamb_8_30_5 = 0.001976745116279;

d_8_30_5 =1.000000000000000e-03;

[AF_8_30_5] = Array_Fact(phi_8_30_5,lamb_8_30_5,d_8_30_5);

phi_8_31 = [1.19275946961384,0.745207247528148,0.289610191793846,-0.126162141802421,-0.574262320592041,-1.03061981959431,-1.44963376022200,-1.89687748839857];

lamb_8_31 =0.001976745116279;

d_8_31 =1.000000000000000e-03;

[AF_8_31] = Array_Fact(phi_8_31,lamb_8_31,d_8_31);

phi_8_31_5 = [1.52756418217238,0.870553260325390,0.199297617408925,-0.444012848329250,-1.11824189670576,-1.76807629779896,-2.43062176996940,-3.09822136558873];

lamb_8_31_5 = 0.001973685139319;

d_8_31_5 =1.000000000000000e-03;

[AF_8_31_5] = Array_Fact(phi_8_31_5,lamb_8_31_5,d_8_31_5);

```
phi_8_32 = [1.87098137489006,0.991335778180549,0.111689067245839,-
0.767955254870435,-1.64760261878812,-
2.52724762927624,2.87629054617840,1.99664612337575];
lamb_8_32 = 0.001973685139319;
d_8_32 =1.000000000000000e-03;
[AF_8_32] = Array_Fact(phi_8_32,lamb_8_32,d_8_32);

phi_8_32_5 =[-0.934937594028017,-2.02109795395887,-
3.13104593989302,2.06011707191353,0.961510390323252,-0.143989898743114,-
1.23932659659586,-2.34592234995301];
lamb_8_32_5 = 0.001961539384615;
d_8_32_5 =1.000000000000000e-03;
[AF_8_32_5] = Array_Fact(phi_8_32_5,lamb_8_32_5,d_8_32_5);

phi_8_33 =[-0.599237410729017,-
1.89132418765237,3.05316169692274,1.73853480949895,0.440844530632130,-
0.913646300936953,-2.21028938884405,2.74615056937435];
lamb_8_33 = 0.001949542201835;
d_8_33 = 1.000000000000000e-03;
[AF_8_33] = Array_Fact(phi_8_33,lamb_8_33,d_8_33);
```

phi_8_33_5 = [-0.250886706802181,-
1.76502759044177,2.95819010285392,1.41515528345640,-0.0957188314223628,-
1.64865594754482,3.06827225087611,1.55741899678799];

lamb_8_33_5 = 0.001940640182648;

d_8_33_5 =1.000000000000000e-03;

[AF_8_33_5] = Array_Fact(phi_8_33_5,lamb_8_33_5,d_8_33_5);

phi_8_34 = [0.105740803852157,-1.63994369776460,2.86445821396914,1.09643173275403,-
0.651572036483785,-2.38586221934003,2.11113792312959,0.336117949135037];

lamb_8_34 = 0.001934750531108;

d_8_34 =1.000000000000000e-03;

[AF_8_34] = Array_Fact(phi_8_34,lamb_8_34,d_8_34);

phi_8_34_5 = [0.460333686962226,-
1.52109152671313,2.77198713895972,0.788022109951291,-
1.19970195949454,3.11367036893811,1.14886524496596,-0.831058470368262];

lamb_8_34_5 = 0.001928896520424;

d_8_34_5 =1.000000000000000e-03;

[AF_8_34_5] = Array_Fact(phi_8_34_5,lamb_8_34_5,d_8_34_5);

phi_8_35 = [0.804211474171522,-1.40201692242000,2.67776538284935,0.479426588912734,-
1.72694713582639,2.34679490305013,0.148944284791511,-2.04188486172405];

lamb_8_35 = 0.001925982779456;

```
d_8_35 =1.000000000000000e-03;

[AF_8_35] = Array_Fact(phi_8_35,lamb_8_35,d_8_35);


[Peak_8_25,I_8_25] = max(AF_8_25);

Angle_8_25=theta(I_8_25);

[Peak_8_25_5,I_8_25_5] = max(AF_8_25_5);

Angle_8_25_5 = theta(I_8_25_5);

[Peak_8_26,I_8_26] = max(AF_8_26);

Angle_8_26=theta(I_8_26);

[Peak_8_26_5,I_8_26_5] = max(AF_8_26_5);

Angle_8_26_5 = theta(I_8_26_5);

[Peak_8_27,I_8_27] = max(AF_8_27);

Angle_8_27=theta(I_8_27);

[Peak_8_27_5,I_8_27_5] = max(AF_8_27_5);

Angle_8_27_5=theta(I_8_27_5);

[Peak_8_28,I_8_28] = max(AF_8_28);

Angle_8_28=theta(I_8_28);

[Peak_8_28_5,I_8_28_5] = max(AF_8_28_5);

Angle_8_28_5=theta(I_8_28_5);

[Peak_8_29,I_8_29] = max(AF_8_29);

Angle_8_29=theta(I_8_29);

[Peak_8_29_5,I_8_29_5] = max(AF_8_29_5);

Angle_8_29_5=theta(I_8_29_5);
```

```
[Peak_8_30,I_8_30] = max(AF_8_30);
Angle_8_30=theta(I_8_30);
[Peak_8_30_5,I_8_30_5] = max(AF_8_30_5);
Angle_8_30_5=theta(I_8_30_5);
[Peak_8_31,I_8_31] = max(AF_8_31);
Angle_8_31=theta(I_8_31);
[Peak_8_31_5,I_8_31_5] = max(AF_8_31_5);
Angle_8_31_5=theta(I_8_31_5);
[Peak_8_32,I_8_32] = max(AF_8_32);
Angle_8_32=theta(I_8_32);
[Peak_8_32_5,I_8_32_5] = max(AF_8_32_5);
Angle_8_32_5=theta(I_8_32_5);
[Peak_8_33,I_8_33] = max(AF_8_33);
Angle_8_33 = theta(I_8_33);
[Peak_8_33_5,I_8_33_5] = max(AF_8_33_5);
Angle_8_33_5=theta(I_8_33_5);
[Peak_8_34,I_8_34] = max(AF_8_34);
Angle_8_34=theta(I_8_34);
[Peak_8_34_5,I_8_34_5] = max(AF_8_34_5);
Angle_8_34_5=theta(I_8_34_5);
[Peak_8_35,I_8_35] = max(AF_8_35);
Angle_8_35=theta(I_8_35);
```

```
Angles_8 = [Angle_8_25,
Angle_8_25_5,Angle_8_26,Angle_8_26_5,Angle_8_27,Angle_8_27_5,Angle_8_28,Angle_8_2
8_5,Angle_8_29,Angle_8_29_5,Angle_8_30,Angle_8_30_5,Angle_8_31,Angle_8_31_5,Angle_
8_32,Angle_8_32_5,Angle_8_33,Angle_8_33_5,Angle_8_34,Angle_8_34_5,Angle_8_35];



%2D plot
figure(4)
plot(theta/pi*180,10*log10(AF_8_25))
hold on
plot(theta/pi*180,10*log10(AF_8_25_5))
plot(theta/pi*180,10*log10(AF_8_26))
plot(theta/pi*180,10*log10(AF_8_26_5))
plot(theta/pi*180,10*log10(AF_8_27))
plot(theta/pi*180,10*log10(AF_8_27_5))
plot(theta/pi*180,10*log10(AF_8_28))
plot(theta/pi*180,10*log10(AF_8_28_5))
plot(theta/pi*180,10*log10(AF_8_29))
plot(theta/pi*180,10*log10(AF_8_29_5))
plot(theta/pi*180,10*log10(AF_8_30))
plot(theta/pi*180,10*log10(AF_8_30_5))
plot(theta/pi*180,10*log10(AF_8_31))
plot(theta/pi*180,10*log10(AF_8_31_5))
```

```
plot(theta/pi*180,10*log10(AF_8_32))
plot(theta/pi*180,10*log10(AF_8_32_5))
plot(theta/pi*180,10*log10(AF_8_33))
plot(theta/pi*180,10*log10(AF_8_33_5))
plot(theta/pi*180,10*log10(AF_8_34))
plot(theta/pi*180,10*log10(AF_8_34_5))
plot(theta/pi*180,10*log10(AF_8_35))
title("8 Element Array Factor: Shift from 25GHz to 35GHz")
ylabel("Array Factor (dB)")
xlabel("Angle (°)")
hold off

figure(5)
polarplot(theta,AF_8_25)
hold on
polarplot(theta,AF_8_25_5)
polarplot(theta,AF_8_26)
polarplot(theta,AF_8_26_5)
polarplot(theta,AF_8_27)
polarplot(theta,AF_8_27_5)
polarplot(theta,AF_8_28)
polarplot(theta,AF_8_28_5)
polarplot(theta,AF_8_29)
```

```
polarplot(theta,AF_8_29_5)
polarplot(theta,AF_8_30)
polarplot(theta,AF_8_30_5)
polarplot(theta,AF_8_31)
polarplot(theta,AF_8_31_5)
polarplot(theta,AF_8_32)
polarplot(theta,AF_8_32_5)
polarplot(theta,AF_8_33)
polarplot(theta,AF_8_33_5)
polarplot(theta,AF_8_34)
polarplot(theta,AF_8_34_5)
polarplot(theta,AF_8_35)
title("8 Element Array Factor: Shift from 25GHz to 35GHz")
hold off

AF_8 = [10*log10(AF_8_25);
10*log10(AF_8_25_5);10*log10(AF_8_26);10*log10(AF_8_26_5);10*log10(AF_8_27);10*log
10(AF_8_27_5);10*log10(AF_8_28);
10*log10(AF_8_28_5);10*log10(AF_8_29);10*log10(AF_8_29_5);10*log10(AF_8_30);10*log
10(AF_8_30_5);10*log10(AF_8_31);
10*log10(AF_8_31_5);10*log10(AF_8_32);10*log10(AF_8_32_5);10*log10(AF_8_33);10*log
10(AF_8_33_5);10*log10(AF_8_34);10*log10(AF_8_34_5);10*log10(AF_8_35)];
```

```
%AF_8 = [AF_8_25; AF_8_25_5;AF_8_26;AF_8_26_5;AF_8_27;AF_8_27_5;AF_8_28;
AF_8_28_5;AF_8_29;AF_8_29_5;AF_8_30;AF_8_30_5;AF_8_31;
AF_8_31_5;AF_8_32;AF_8_32_5;AF_8_33;AF_8_33_5;AF_8_34;AF_8_34_5;AF_8_35];

figure(13)
surf(theta/pi*180,F,AF_8,'FaceAlpha',0.5)
hold on
title('8 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off

figure(69)
mesh(theta/pi*180,F,AF_8)
hold on
title('8 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off
%% 16 Element
```

```
phi_16_25 = [1.26278105113276,-2.82300813991502,-
0.621822060704180,1.57493548412682,-2.50685375310158,-
0.311206287378631,1.89114652747666,-
2.19395762936831,0.0066382154147763 3,2.20189310199877,-
1.87881160245706,0.318584513674744,2.51977059288558,-
1.56665716946300,0.634738900488186,2.83038636621113];
d_16_25 = 1.000000000000000e-03;
lamb_16_25 = 0.002066451539708;
[AF_16_25] = Array_Fact(phi_16_25,lamb_16_25,d_16_25);
[Peak_16_25,I_16_25] = max(AF_16_25);
Angle_16_25=theta(I_16_25);

phi_16_25_5 = [1.59636025711395,-2.71053536012887,-
0.729736366961039,1.25015953216549,-3.05722419858464,-
1.07427358796196,0.901229669258620,2.88534775484502,-
1.42308011951155,0.558405429498390,2.53656574584162,-
1.76833028276828,0.215410050746540,2.18838768185994,-2.11005558519151,-
0.135854662613510];
d_16_25_5 = 1.000000000000000e-03;
lamb_16_25_5 = 0.002059774798061;
[AF_16_25_5] = Array_Fact(phi_16_25_5,lamb_16_25_5,d_16_25_5);
[Peak_16_25_5,I_16_25_5] = max(AF_16_25_5);
Angle_16_25_5 = theta(I_16_25_5);
```

```
phi_16_26 =[1.93063720835130,-2.60112457006603,-
0.838493084980375,0.927715715941032,2.67177150734658,-1.84140506908808,-
0.0816408778428250,1.66973223069129,-2.84248596336802,-
1.09234839709398,0.664821910451594,2.43448495786850,-2.09618932421462,-
0.337906877268059,1.42877027654676,-3.10723749558711];
d_16_26 = 1.000000000000000e-03;
lamb_16_26 = 0.002049840192926;
[AF_16_26] = Array_Fact(phi_16_26,lamb_16_26,d_16_26);
[Peak_16_26,I_16_26] = max(AF_16_26);
Angle_16_26=theta(I_16_26);

phi_16_26_5 = [2.25885343544798,-2.49065052613938,-
0.948752229261120,0.599349732578323,2.12787098936081,-2.61794387135193,-
1.06969612151306,0.471650463229098,1.99964367785226,-2.73884553807324,-
1.19482119917627,0.339115664779357,1.87599810175320,-2.85798777351808,-
1.32377123786690,0.208349997026537];
d_16_26_5 = 1.000000000000000e-03;
lamb_16_26_5 = 0.002043270192308;
[AF_16_26_5] = Array_Fact(phi_16_26_5,lamb_16_26_5,d_16_26_5);
[Peak_16_26_5,I_16_26_5] = max(AF_16_26_5);
Angle_16_26_5 = theta(I_16_26_5);
```

```
phi_16_27 =[2.58666285720566,-2.37755859873881,-
1.05774175944389,0.263196423465908,1.58225250600419,2.90025388003020,-
2.06341862947975,-0.742949764317164,0.577670644307798,1.89572295689573,-
3.06909867239340,-1.74895104086744,-
0.428142470177051,0.891661578893625,2.20914530596676,-2.75489912169955];
d_16_27 = 1.000000000000000e-03;
lamb_16_27 = 0.002040000960000;
[AF_16_27] = Array_Fact(phi_16_27,lamb_16_27,d_16_27);
[Peak_16_27,I_16_27] = max(AF_16_27);
Angle_16_27=theta(I_16_27);

phi_16_27_5 = [2.91589482627884,-2.26771208600955,-1.16804175128642,-
0.0680254207405426,1.03186532516471,2.13119658543130,-3.05260882404889,-
1.95336124026603,-
0.854163559544963,0.245718437320111,1.34575436910165,2.44556265968536,-
2.73799252654865,-1.63905051123053,-0.540109409958962,0.559655849791804];
d_16_27_5 = 1.000000000000000e-03;
lamb_16_27_5 = 0.002033493779904;
[AF_16_27_5] = Array_Fact(phi_16_27_5,lamb_16_27_5,d_16_27_5);
[Peak_16_27_5,I_16_27_5] = max(AF_16_27_5);
Angle_16_27_5=theta(I_16_27_5);
```

```
phi_16_28 = [-3.03787251712466,-2.15822655626264,-1.27858000421308,-
0.398933803920224,0.480711929650086,1.36035836065315,2.24000507437004,3.1196510593
7486,-2.28388879777473,-1.40424300553096,-
0.524597354715060,0.355048073122249,1.23469383080261,2.11433989575037,2.9939855144
1885,-2.40955389977892];
d_16_28 = 1.000000000000000e-03;
lamb_16_28 = 0.002027027980922;
[AF_16_28] = Array_Fact(phi_16_28,lamb_16_28,d_16_28);
[Peak_16_28,I_16_28] = max(AF_16_28);
Angle_16_28=theta(I_16_28);

phi_16_28_5 =[-2.70852317956127,-2.04894721275165,-1.38920463019641,-
0.729338131357553,-
0.0695582061944172,0.590059747605416,1.24966397783605,1.90940274957275,2.569225160
34105,-3.05421142847848,-2.39457769824718,-1.73494627416312,-1.07522118054210,-
0.415443201951348,0.244287178560418,0.903947377834660];
d_16_28_5 = 1.000000000000000e-03;
lamb_16_28_5 =0.002020603169572;
[AF_16_28_5] = Array_Fact(phi_16_28_5,lamb_16_28_5,d_16_28_5);
[Peak_16_28_5,I_16_28_5] = max(AF_16_28_5);
Angle_16_28_5=theta(I_16_28_5);
```

phi_16_29 =

[0.759310447233388,1.19898592219790,1.63846863461263,2.07736859402457,2.5167010818

3167,2.95768250309447,-2.88318844942611,-2.44127449820376,-2.00093964873122,-

1.56153118653167,-1.12161173019829,-0.680882829887696,-

0.240491932239408,0.198583613185746,0.636800165861997,1.07530221656169];

d_16_29 = 1.000000000000000e-03;

lamb_16_29 = 0.002011041955836;

[AF_16_29] = Array_Fact(phi_16_29,lamb_16_29,d_16_29);

[Peak_16_29,I_16_29] = max(AF_16_29);

Angle_16_29=theta(I_16_29);

phi_16_29_5

=[1.09028882364805,1.30877325098899,1.52821359141791,1.74844618553886,1.9695102153

3575,2.19118099633241,2.41232078402444,2.63156947560546,2.84900562074407,3.06660443

718530,-2.99669472111197,-2.77440242343005,-2.55169340620384,-2.33063502973843,-

2.11159273654471,-1.89347610081150];

d_16_29_5 = 1.000000000000000e-03;

lamb_16_29_5 = 0.002004717924528;

[AF_16_29_5] = Array_Fact(phi_16_29_5,lamb_16_29_5,d_16_29_5);

[Peak_16_29_5,I_16_29_5] = max(AF_16_29_5);

Angle_16_29_5=theta(I_16_29_5);

phi_16_30

=[1.42102433881405,1.41901667294220,1.41565595239571,1.42075297064464,1.4253042871

8987,1.42102436624400,1.41901664550077,1.41565597976823,1.42075294335954,1.42530431

450699,1.42102433881403,1.41901667294218,1.41565595239569,1.42075297064463,1.425304

28718985,1.42102436624398];

d_16_30 = 1.000000000000000e-03;

lamb_16_30 = 0.001998433542320;

[AF_16_30] = Array_Fact(phi_16_30,lamb_16_30,d_16_30);

[Peak_16_30,I_16_30] = max(AF_16_30);

Angle_16_30=theta(I_16_30);

phi_16_30_5

=[1.75149768816596,1.52988933805453,1.30961121609045,1.09198952743795,0.8718049711

69765,0.649348674788147,0.430346519853735,0.213045993535701,-0.00838224626646121,-

0.231049176605717,-0.449423969981092,-0.667164713990159,-0.888850416492290,-

1.11023465157904,-1.32816150087431,-1.54731146228054];

d_16_30_5 = 1.000000000000000e-03;

lamb_16_30_5 = 0.001992188437500;

[AF_16_30_5] = Array_Fact(phi_16_30_5,lamb_16_30_5,d_16_30_5);

[Peak_16_30_5,I_16_30_5] = max(AF_16_30_5);

Angle_16_30_5=theta(I_16_30_5);

```
phi_16_31=[2.08223101344503,1.64155983138690,1.20151876163556,0.763474593847976,0.3
21745629929670,-0.118547484551293,-0.556315447798092,-0.997606199607055,-
1.43759103860314,-1.87584350538791,-2.31735735513027,-
2.75672230676587,3.08827071655415,2.64620055409184,2.20645860999101,1.768189236464
50];
d_16_31 = 1.000000000000000e-03;
lamb_16_31 = 0.001985982242991;
[AF_16_31] = Array_Fact(phi_16_31,lamb_16_31,d_16_31);
[Peak_16_31,I_16_31] = max(AF_16_31);
Angle_16_31=theta(I_16_31);

phi_16_31_5
=[2.41402331065685,1.75418825380569,1.09437566385687,0.435339584193381,-
0.225774757784126,-0.884514209932815,-1.54463332456229,-2.20411477531888,-
2.86346130077784,2.75872356987010,2.10009819403607,1.43974982708380,0.780583831820
940,0.120877578222097,-0.539901321781209,-1.19851971936518];
d_16_31_5 = 1.000000000000000e-03;
lamb_16_31_5 = 0.001979814596273;
[AF_16_31_5] = Array_Fact(phi_16_31_5,lamb_16_31_5,d_16_31_5);
[Peak_16_31_5,I_16_31_5] = max(AF_16_31_5);
Angle_16_31_5=theta(I_16_31_5);
```

```
phi_16_32 =
[2.74756916379502,1.86792363481630,0.988277071385303,0.108631717409989,-
0.771014962823882,-1.65066018449387,-
2.53030718452739,2.87323392092059,1.99358646051594,1.11394142150068,0.234294005858
352,-0.645349712727285,-1.52499740255203,-
2.40464210185246,2.99889567452992,2.11925150991633];
d_16_32 = 1.000000000000000e-03;
lamb_16_32 = 0.001973685139319;
[AF_16_32] = Array_Fact(phi_16_32,lamb_16_32,d_16_32);
[Peak_16_32,I_16_32] = max(AF_16_32);
Angle_16_32=theta(I_16_32);

phi_16_32_5 =[3.08315379448942,1.98290256965515,0.883319115023272,-
0.215830437772242,-1.31570431207285,-
2.41464134046388,2.76794431903955,1.66939900725703,0.568710308361355,-
0.529449441896678,-1.62998712110122,-
2.72922199561207,2.45426135676565,1.35456190557540,0.255655468879783,-
0.844581465443658];
d_16_32_5 = 1.000000000000000e-03;
lamb_16_32_5 = 0.001967593518519;
[AF_16_32_5] = Array_Fact(phi_16_32_5,lamb_16_32_5,d_16_32_5);
[Peak_16_32_5,I_16_32_5] = max(AF_16_32_5);
Angle_16_32_5=theta(I_16_32_5);
```

```
phi_16_33 = [0.273971030558906,-1.03994466955450,-
2.35892235082433,2.59917002117957,1.28595669227891,-0.0393014521839654,-
1.35598393219180,-2.67131011207386,2.28434278705870,0.971035843112990,-
0.351733652539646,-1.67180447828975,-
2.98487827249100,1.97108321023124,0.654942919689688,-0.663981737715697];
d_16_33 = 1.000000000000000e-03;
lamb_16_33 = 0.001958526267281;
[AF_16_33] = Array_Fact(phi_16_33,lamb_16_33,d_16_33);
[Peak_16_33,I_16_33] = max(AF_16_33);
Angle_16_33 = theta(I_16_33);

phi_16_33_5 = [0.613170203344584,-0.921415675739363,-
2.46012585347110,2.27770650039815,0.742210435443507,-0.796350487452033,-
2.33867804281439,2.40658215378614,0.870748642334340,-0.674906544420974,-
2.21314259761955,2.53528116016130,0.993799678412218,-0.549294179387401,-
2.08377474087795,2.65798187687486];
d_16_33_5 = 1.000000000000000e-03;
lamb_16_33_5 = 0.001952527718224;
[AF_16_33_5] = Array_Fact(phi_16_33_5,lamb_16_33_5,d_16_33_5);
[Peak_16_33_5,I_16_33_5] = max(AF_16_33_5);
Angle_16_33_5=theta(I_16_33_5);
```

```
phi_16_34 = [0.954667121859333,-0.801393259644497,-
2.55999641472852,1.95943669817393,0.199790920295854,-
1.55620972736159,2.96887604658275,1.20707706432523,-0.554373885568348,-
2.31055765837989,2.21413834086137,0.453444044018335,-1.30776477879824,-
3.06479776489700,1.46048422398182,-0.301254599035611];
d_16_34 = 1.000000000000000e-03;
lamb_16_34 = 0.001946565801527;
[AF_16_34] = Array_Fact(phi_16_34,lamb_16_34,d_16_34);
[Peak_16_34,I_16_34] = max(AF_16_34);
Angle_16_34=theta(I_16_34);

phi_16_34_5 = [1.29771513647938,-0.679814731769685,-
2.65845008342612,1.64316745296346,-0.338074868720102,-
2.31640416522934,1.98931277938964,0.0118552843970693,-
1.96795950297836,2.33340508561456,0.353578012546350,-
1.62477618428149,2.68080773072899,0.702845931293683,-
1.27777512412675,3.02477996751944];
d_16_34_5 = 1.000000000000000e-03;
lamb_16_34_5 = 0.001940640182648;
[AF_16_34_5] = Array_Fact(phi_16_34_5,lamb_16_34_5,d_16_34_5);
[Peak_16_34_5,I_16_34_5] = max(AF_16_34_5);
Angle_16_34_5=theta(I_16_34_5);
```

```
phi_16_35 =[1.64183926789448,-0.556625009693853,-2.75537638207658,1.32788477285450,-
0.872463477534371,-3.07186640222864,1.01233031737814,-
1.18700216808767,2.89719087691609,0.698740345095319,-
1.49975338569531,2.58496764389595,0.386216271513222,-
1.81370788073528,2.26912917605543,0.0697262513611558];
d_16_35 = 1.000000000000000e-03;
lamb_16_35 = 0.001934750531108;
[AF_16_35] = Array_Fact(phi_16_35,lamb_16_35,d_16_35);
[Peak_16_35,I_16_35] = max(AF_16_35);
Angle_16_35=theta(I_16_35);

Angles_16 = [Angle_16_25,
Angle_16_25_5,Angle_16_26,Angle_16_26_5,Angle_16_27,Angle_16_27_5,Angle_16_28,Ang
le_16_28_5,Angle_16_29,Angle_16_29_5,Angle_16_30,Angle_16_30_5,Angle_16_31,Angle_1
6_31_5,Angle_16_32,Angle_16_32_5,Angle_16_33,Angle_16_33_5,Angle_16_34,Angle_16_3
4_5,Angle_16_35];



figure(6)
plot(theta/pi*180,10*log10(AF_16_25))
hold on
plot(theta/pi*180,10*log10(AF_16_25_5))
```

```
plot(theta/pi*180,10*log10(AF_16_26))
plot(theta/pi*180,10*log10(AF_16_26_5))
plot(theta/pi*180,10*log10(AF_16_27))
plot(theta/pi*180,10*log10(AF_16_27_5))
plot(theta/pi*180,10*log10(AF_16_28))
plot(theta/pi*180,10*log10(AF_16_28_5))
plot(theta/pi*180,10*log10(AF_16_29))
plot(theta/pi*180,10*log10(AF_16_29_5))
plot(theta/pi*180,10*log10(AF_16_30))
plot(theta/pi*180,10*log10(AF_16_30_5))
plot(theta/pi*180,10*log10(AF_16_31))
plot(theta/pi*180,10*log10(AF_16_31_5))
plot(theta/pi*180,10*log10(AF_16_32))
plot(theta/pi*180,10*log10(AF_16_32_5))
plot(theta/pi*180,10*log10(AF_16_33))
plot(theta/pi*180,10*log10(AF_16_33_5))
plot(theta/pi*180,10*log10(AF_16_34))
plot(theta/pi*180,10*log10(AF_16_34_5))
plot(theta/pi*180,10*log10(AF_16_35))
title("16 Element Array Factor: Shift from 25GHz to 35GHz")
ylabel("Array Factor (dB)")
xlabel("Angle (°)")
hold off
```

```
figure(7)
polarplot(theta,AF_16_25)
hold on
polarplot(theta,AF_16_25_5)
polarplot(theta,AF_16_26)
polarplot(theta,AF_16_26_5)
polarplot(theta,AF_16_27)
polarplot(theta,AF_16_27_5)
polarplot(theta,AF_16_28)
polarplot(theta,AF_16_28_5)
polarplot(theta,AF_16_29)
polarplot(theta,AF_16_29_5)
polarplot(theta,AF_16_30)
polarplot(theta,AF_16_30_5)
polarplot(theta,AF_16_31)
polarplot(theta,AF_16_31_5)
polarplot(theta,AF_16_32)
polarplot(theta,AF_16_32_5)
polarplot(theta,AF_16_33)
polarplot(theta,AF_16_33_5)
polarplot(theta,AF_16_34)
polarplot(theta,AF_16_34_5)
```

```
polarplot(theta,AF_16_35)
title("16 Element Array Factor: Shift from 25GHz to 35GHz")
hold off

AF_16 = [10*log10(AF_16_25);
10*log10(AF_16_25_5);10*log10(AF_16_26);10*log10(AF_16_26_5);10*log10(AF_16_27);10
*log10(AF_16_27_5);10*log10(AF_16_28);
10*log10(AF_16_28_5);10*log10(AF_16_29);10*log10(AF_16_29_5);10*log10(AF_16_30);10
*log10(AF_16_30_5);10*log10(AF_16_31);
10*log10(AF_16_31_5);10*log10(AF_16_32);10*log10(AF_16_32_5);10*log10(AF_16_33);10
*log10(AF_16_33_5);10*log10(AF_16_34);10*log10(AF_16_34_5);10*log10(AF_16_35)];

%AF_16 = [AF_16_25;
AF_16_25_5;AF_16_26;AF_16_26_5;AF_16_27;AF_16_27_5;AF_16_28;
AF_16_28_5;AF_16_29;AF_16_29_5;AF_16_30;AF_16_30_5;AF_16_31;
AF_16_31_5;AF_16_32;AF_16_32_5;AF_16_33;AF_16_33_5;AF_16_34;AF_16_34_5;AF_16
_35];
figure(14)
surf(theta/pi*180,F,AF_16,'FaceAlpha',0.5)
hold on
title('16 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
```

```
zlabel('Array Factor (dB)')
hold off

figure(73)
mesh(theta/pi*180,F,AF_16)
hold on
title('16 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off



%% 32 Element
phi_32_25 = [-1.09415337815110,2.98991536254352,0.790802000368829,-
1.40831403551174,2.67575799688373,0.476640965099413,-
1.72247215877795,2.36159719741899,0.162483741978829,-
2.03663296187398,2.04743927543872,-0.151677291046243,-
2.35079065322093,1.73327861807809,-0.465834656706031,-
2.66495168849035,1.41912049481188,-0.779995456170770,-
2.97910891161093,1.10495969171584,-
1.09415337815104,2.98991536254358,0.790802000368892,-
1.40831403551168,0.476640965099476,2.67575799688379,-
```

1.72247215877788,2.36159719741905,0.162483741978892,-

2.03663296187392,2.04743927543878,-0.151677291046181];

d_32_25 = 1.000000000000000e-03;

lamb_32_25 =0.002068966490872;

[AF_32_25] = Array_Fact(fliplr(phi_32_25),lamb_32_25,d_32_25);

phi_32_25_5 = [-2.94362801763378,1.35817546631419,-0.619111668168191,-

2.60013215398399,1.70469965620949,-0.273694371708683,-

2.25456451888925,2.05144970788599,0.0697468820479341,-

1.90741208344683,2.39575137450167,0.416752624783334,-

1.56219762346282,2.74054186859446,0.763037446611102,-

1.21772616714511,3.08814845747454,1.10698932086726,-0.871297283825865,-

2.85040258463150,1.45277626254379,-0.524419952521796,-

2.50639568393278,1.79978346790862,-2.15923601418053,-

0.180859727774810,2.14519915311890,0.164472812605775,-

1.81318568961368,2.48915026174480,0.511933152287472,-1.46875960805431];

d_32_25_5 = 1.000000000000000e-03;

lamb_32_25_5 = 0.002062690556117;

[AF_32_25_5] = Array_Fact(fliplr(phi_32_25_5),lamb_32_25_5,d_32_25_5);

phi_32_26 = [-1.65464174511567,2.86824338666554,1.11004484608812,-

0.649454357671906,-2.40933021602707,2.11523173314375,0.355536363095480,-

1.40398310248831,3.12091739092123,1.36069386249323,-0.398612505132398,-

2.15713357540075,2.36571213439475,0.607353451076892,-1.15205276424328,-2.91220181216059,1.61258367251481,-0.146940965020143,-1.90655989560313,2.61807104083003,0.858084992564689,-0.901413873867872,-2.65961556569969,1.86324535705166,-1.65464177254619,0.104401193359109,2.86824341409886,1.11004481866502,-0.649454330237255,-2.40933024346644,2.11523176057744,0.355536335662979];

d_32_26 = 1.000000000000000e-03;

lamb_32_26 = 0.002054381631420;

[AF_32_26] = Array_Fact(fliplr(phi_32_26),lamb_32_26,d_32_26);

phi_32_26_5 = [2.77589576286695,1.23568605547085,-0.303043668145454,-1.84225078556871,2.90089737549871,1.36162289893910,-0.176876134514775,-1.71692481380839,3.02627494528216,1.48770141303309,-0.0513147215578458,-1.59157840597318,-3.13087120329328,1.61361737334880,0.0737097684613204,-1.46595720414559,-3.00457505496240,1.73919917893793,0.198828693352062,-1.34000449816763,-2.87869222375717,1.86450964520970,0.324567791405619,-1.21390840294017,1.98983465068929,-2.75345063931944,0.450803480675044,-1.08801084759565,-2.62841662447721,2.11550109767881,0.576927352579356,-0.962548923153032];

d_32_26_5 = 1.000000000000000e-03;

lamb_32_26_5 = 0.002048193734940;

[AF_32_26_5] = Array_Fact(fliplr(phi_32_26_5),lamb_32_26_5,d_32_26_5);

```
phi_32_27 =[0.923473521574951,-0.391636440317976,-1.71699787955282,-
3.04158402687037,1.92582318096972,0.611542719799459,-0.706657115360654,-
2.03420348063097,2.92805871253594,1.61386407166891,0.298584833481828,-
1.02323647403888,-2.35028440046174,2.61594049991049,1.30160624731483,-
0.0158639884396555,-1.34044630271861,-
2.66451087097878,2.30446873126841,0.988326931555631,-0.331702146545003,-
1.65710716565550,-2.97687251557418,1.99262779561323,-
0.648289185265363,0.673684957508538,-
1.97225848490354,2.99512388269678,1.67953759772237,0.357857745138845,-
0.964688952099392,-2.28553264684677];
d_32_27 = 1.000000000000000e-03;
lamb_32_27 = 0.002044089138277;
[AF_32_27] = Array_Fact(fliplr(phi_32_27),lamb_32_27,d_32_27);

phi_32_27_5 =[2.20974782121382,1.11019013739962,0.0106326567042710,-
1.08892521186212,-
2.18848354809873,2.99514466935010,1.89558806268004,0.796030993380177,-
0.303525977407109,-1.40308424835065,-
2.50264282644653,2.68098527867659,1.58142837509658,0.481871563346380,-
0.617685178014648,-1.71724340265165,-
2.81680157212407,2.36682631034973,1.26726871055223,0.167711794167490,-
0.931844832375988,-2.03140251619018,-3.13095999688553,2.05266744172766,-
0.146447984239709,0.953109105491053,-1.24600459090976,-
```

2.34556166020963,2.83806667618267,1.73850840523913,0.638949827143255,-
0.460607374913213];
d_32_27_5 = 1.000000000000000e-03;
lamb_32_27_5 = 0.002033899262213;
[AF_32_27_5] = Array_Fact(fliplr(phi_32_27_5),lamb_32_27_5,d_32_27_5);

phi_32_28_1 = [0.356487800807601,-0.522933029478595,-1.40328736691838,-
2.28384411086515,3.11954957985041,2.23990578286917,1.35970099442510,0.479679211454
935,-0.399566796135495,-1.27860429756669,-2.15772684416312,-
3.03675513909629,2.36705959248713,1.48709454564541,0.607291778866970,-
0.272198719912767,-1.15234800056408,-2.03310664869256,-
2.91321738217148,2.49065020188711,1.61123448026698,0.731723163772598,-
0.147443056030348,-1.02660003137662,-2.78510488019898,-
1.90593650898023,2.61865965154719,1.73830525921549,0.857748570158046,-
0.0220431011476003,-0.901686843292978,-1.78189168661608];
phi_32_28 = [0.359390807337732,-0.519457993768241,-1.40158293128756,-
2.28442890965814,3.11859696591313,2.23895504846952,1.35734882748110,0.476394538974
989,-0.401844834608989,-1.27936259298550,-2.15717833694312,-
3.03466381320964,2.36984592151105,1.48908797088953,0.608891540060913,-
0.270210517240080,-1.15162274039727,-2.03517852232504,-
2.91644816805926,2.48824492764186,1.60940597239000,0.730234988328556,-
0.147731845486835,-1.02566704912986,-2.78220187362936,-

1.90422853348435,2.62213468726568,1.74000969478778,0.857163771366919,-0.0229957150238152,-0.902637577705060,-1.78424385360718];

d_32_28 = 1.000000000000000e-03;

lamb_32_28 = 0.002029851701493;

[AF_32_28] = Array_Fact(fliplr(phi_32_28_1),lamb_32_28,d_32_28);

phi_32_28_5 = [1.64267919639329,0.983042079017854,0.323405610319378,-0.336385758419156,-0.996265144851189,-1.65605771902355,-2.31572714781786,-2.97540651368256,2.64798609602220,1.98813704939631,1.32834995462434,0.668642472583032,0.00892932641291094,-0.650850928147301,-1.31066766871976,-1.97045819388500,-2.63020552618465,2.99324508051525,2.33349003827886,1.67370418503437,1.01390186722719,0.354116585578481,-0.305625107772701,-0.965344857662239,-2.28492559055551,-1.62510406038295,-2.94474297955575,2.67870867528192,2.01903104364786,1.35929142624210,0.699445364762354,0.0396046478331804];

d_32_28_5 = 1.000000000000000e-03;

lamb_32_28_5 = 0.002029851701493;

[AF_32_28_5] = Array_Fact(fliplr(phi_32_28_5),lamb_32_28_5,d_32_28_5);

phi_32_29 = [-0.211833127427026,-0.651456380052641,-1.09135489693539,-1.53132312401514,-1.97119014686007,-2.41102384304500,-2.85098830150338,2.99210345088465,2.55205647637383,2.11215843494495,1.67230237942257,1.23235071219665,0.792377318324776,0.352611373017664,-0.0868680389022195,-

0.526276863996463,-0.965895019465963,-1.40574756452495,-1.84560828113582,-2.28531549962533,-2.72497845382014,3.11836949073980,2.67832204182072,2.23823882067100,1.35830104499486,1.79825400352451,0.918265942105030,0.478218086188411,0.0383670902405309,-0.401218672349248,-0.840738423756113,-1.28043231965453];

d_32_29 = 1.000000000000000e-03;

lamb_32_29 = 0.002013821283317;

[AF_32_29] = Array_Fact(fliplr(phi_32_29),lamb_32_29,d_32_29);

phi_32_29_5 = [-2.06624612810584,-2.28958629853223,-2.51199397977118,-2.73093823545187,-2.94639718884729,3.12152351098541,2.90275522948331,2.67994914382682,2.45598184792097,2.23400052193949,2.01471536430715,1.79677685905491,1.57884642560966,1.36043465040306,1.14099190025391,0.919532466734930,0.696053529257863,0.472801865141623,0.252875371644007,0.0369599884135575,-0.177914755650867,-0.395701605481893,-0.617583902348870,-0.841294498736565,-1.28432122664744,-1.06388566203029,-1.50327141245063,-1.72144794600647,-1.93918256415926,-2.15744852796165,-2.37794313768972,-2.60121902309564];

d_32_29_5 = 1.000000000000000e-03;

lamb_32_29_5 = 0.002009853162562;

[AF_32_29_5] = Array_Fact(fliplr(phi_32_29_5),lamb_32_29_5,d_32_29_5);

```
phi_32_30 = [-0.778820756835124,-0.778821730829277,-0.778821752229104,-
0.778820863279061,-0.778820220658878,-0.778820784267737,-0.778821703396655,-
0.778821779661701,-0.778820835846432,-0.778820248091495,-0.778820756835102,-
0.778821730829257,-0.778821752229084,-0.778820863279039,-0.778820220658856,-
0.778820784267717,-0.778821703396634,-0.778821779661679,-0.778820835846410,-
0.778820248091475,-0.778820756835081,-0.778821730829234,-0.778821752229062,-
0.778820863279018,-0.778820784267694,-0.778820220658836,-0.778821703396612,-
0.778821779661659,-0.778820835846389,-0.778820248091452,-0.778820756835060,-
0.778821730829214];
d_32_30 = 1.000000000000000e-03;
lamb_32_30 = 0.002000000941176;
[AF_32_30] = Array_Fact(fliplr(phi_32_30),lamb_32_30,d_32_30);

phi_32_30_5 = [-2.62800386345566,-2.41377430393208,-2.19628700777810,-
1.97096741285172,-1.75031112544057,-1.53485930461771,-1.31299646646849,-
1.08848802273599,-0.872215777539216,-0.656352683382928,-0.431231101685723,-
0.207880851298056,0.0054889922339324 2,0.222006591490562,0.448108362868604,0.669952
538388323,0.883922159721312,1.10336409930591,1.32864225614660,1.54750200906206,1.76
352238027539,1.98704562963783,2.21047069431428,2.42522052744842,2.86849674385752,2.
64234336365392,3.09016023705741,-2.98034305968822,-2.76198826417515,-
2.53557538010271,-2.31550906900884,-2.10122056641226];
d_32_30_5 = 1.000000000000000e-03;
lamb_32_30_5 = 0.001996087045010;
```

[AF_32_30_5] = Array_Fact(fliplr(phi_32_30_5),lamb_32_30_5,d_32_30_5);

phi_32_31 =[-1.34209744687622,-0.903091269774678,-0.463241277765023,-0.0223149633099917,0.416653025094194,0.856305145014887,1.29693928389554,1.73594509071061,2.17591605301383,2.61644133567386,3.05521569416214,-2.78785038076748,-2.34712166507028,-1.90820945741428,-1.46821828670870,-1.02801648447521,-0.589206687455122,-0.148698365548373,0.291822983907342,0.730587237676591,1.17078489364073,1.61091832307646,2.04984016538429,2.49042605146097,-2.91393030913989,2.93048233071311,-2.47330487248003,-2.03318275771496,-1.59424577700127,-1.15376081638327,-0.714103409619154,-0.275106167793659];
d_32_31 = 1.000000000000000e-03;
lamb_32_31 = 0.001986368997079;
[AF_32_31] = Array_Fact(fliplr(phi_32_31),lamb_32_31,d_32_31);

phi_32_31_5 = [3.09016789234808,-2.53388477747353,-1.87437677074132,-1.21396271534152,-0.555155984408524,0.105734518927658,0.764611230218103,1.42450053205460,2.08459987337025,2.74353956359630,-2.87874684679248,-2.22004091729174,-1.55980151654291,-0.900064010720901,-0.240904281282914,0.419905918919164,1.07854312096430,1.73906689955391,2.39843111163733,3.05787622953465,-2.56467575719320,-1.90599871651514,-1.24528395440424,-

0.586268164129453,0.733885368161744,0.0734999951270385,1.39270714657990,2.05350054774905,2.71222708421201,-2.91086118625145,-2.25077032935215,-1.59171326432219];

d_32_31_5 = 1.000000000000000e-03;

lamb_32_31_5 = 0.001980583456311;

[AF_32_31_5] = Array_Fact(fliplr(phi_32_31_5),lamb_32_31_5,d_32_31_5);

phi_32_32 = [1.23976837190270,2.12027564081434,2.99932287441566,-2.40548572835337,-1.52429612290464,-0.645227522451134,0.235837350190882,1.11274443668382,1.99493630814334,2.872020997468802,-2.52838355485773,-1.65166833719258,-0.768812722457205,0.106633108580703,0.989864690198307,1.86697377086453,2.75017007968187,-2.65764140334401,-1.77509445412323,-0.897704943926987,-0.0151310348340627,0.861930147692106,1.74346206950633,2.62126300194233,-1.90182430911477,-2.78122820670652,-1.02131698532276,-0.142269806616633,0.736106952670458,1.61729650326018,2.49636515854952,-2.90575533082988];

d_32_32 = 1.000000000000000e-03;

lamb_32_32 =0.001974831519845;

[AF_32_32] = Array_Fact(fliplr(phi_32_32),lamb_32_32,d_32_32);

phi_32_32_5 =[2.53317407751696,-2.65045473274255,-1.55089689840741,-0.451338572447899,0.648217344332053,1.74777614967900,2.84733204264573,-2.33629436217382,-1.23673787289837,-

0.137179387235844,0.962376613555153,2.06193483637302,-3.12169292379670,-
2.02213607436059,-
0.922577317357839,0.176978159734816,1.27653709623544,2.37609346548881,-
2.80753304584118,-1.70797696745222,-
0.608418576072909,0.491137920847171,1.59069575518231,2.69025408114182,-
1.39381650391087,-2.49337530925781,-
0.294260610944142,0.805298291415896,1.90485478069135,3.00441326635388,-
2.17921604003471,-1.07965781721685];
d_32_32_5 = 1.000000000000000e-03;
lamb_32_32_5 =0.001967214040501;
[AF_32_32_5] = Array_Fact(fliplr(phi_32_32_5),lamb_32_32_5,d_32_32_5);

phi_32_33 = [0.687470797737882,2.00600468461709,-2.95847112280538,-
1.63758539994715,-0.320026268792876,1.00109261513689,2.32094351813581,-
2.64492273536569,-1.32328770920522,-
0.00549390074473338,1.31460239586157,2.63565353753824,-2.33090908880401,-
1.00945995744104,0.309268675422104,1.62824747326213,2.94992450172712,-
2.01637360819621,-0.695974851521637,0.624030414650355,1.94222418529801,-
3.01947073802097,-1.70145921110077,-
0.382575479291668,2.25660270995080,0.938560792808155,-2.70601990692333,-
1.38645506508565,-0.0689723609661003,1.25271815257514,2.57129990046519,-
2.39263839956503];
d_32_33 = 1.000000000000000e-03;

```
lamb_32_33 = 0.001961539384615;

[AF_32_33] = Array_Fact(fliplr(phi_32_33),lamb_32_33,d_32_33);


phi_32_33_5 =[1.98332526245908,-2.76124013327083,-
1.22197279649743,0.318632334457005,1.85728051834055,-2.88716861325208,-
1.34696851727060,0.192495225815338,1.73111884648943,-3.01192468895595,-
1.47251113225972,0.0658738355697341,1.60584720512598,-3.13715374411287,-
1.59876242942826,-0.0595584643286369,1.48069869616747,3.01950585611105,-
1.72448146170737,-0.184254664017230,1.35467073863597,2.89318211914078,-
1.84948814807069,-0.309974019295847,2.76817645648147,1.22851182046153,-
1.97504842476189,-0.436616562545165,1.10328292140659,2.64324506142094,-
2.10140363379867,-0.562452991738305];
d_32_33_5 = 1.000000000000000e-03;
lamb_32_33_5 = 0.001954023908046;
[AF_32_33_5] = Array_Fact(fliplr(phi_32_33_5),lamb_32_33_5,d_32_33_5);


phi_32_34 =[0.140047758764659,1.89868826798599,-2.62567188589740,-
0.865586743577467,0.893842953978754,2.65276125549776,-1.87156359388724,-
0.111656913682276,1.64791595598424,-2.87662195254492,-
1.11753752574581,0.642223535976324,2.40214407236731,-2.12258927871164,-
0.363750395918602,1.39605750418231,-3.12702986230103,-
1.36818255555402,0.390089450181599,2.14982851378125,-2.37326792451397,-
0.613861781132128,1.14430755272072,2.90358983722365,0.140047731344475,-
```

1.61949194066903,1.89868829542553,-2.62567191333650,-
0.865586716141180,0.893842926552350,2.65276128291998,-1.87156362132703];
d_32_34 = 1.000000000000000e-03;
lamb_32_34 = 0.001948424985673;
[AF_32_34] = Array_Fact(fliplr(phi_32_34),lamb_32_34,d_32_34);

phi_32_34_5 = [-1.70244867099323,0.278627646311367,2.25828374416867,-
2.04799059808343,-0.0700718312086037,1.90949828897338,-2.39337824960068,-
0.412467635547045,1.56627885977713,-2.73918321340512,-
0.760523915289807,1.21828486582961,-3.08451232667345,-
1.10397176703689,0.874899948257956,2.85358484926307,-
1.45179013305664,0.526724877699022,2.50754826292397,-
1.79522697581291,0.184398682291518,2.16254211592018,-2.14381507258403,-
0.164451461931367,-2.48586124931003,1.81662297004230,-
0.506329123751478,1.47039774780720,-2.83543301794570,-
0.855111606669645,1.12591639906268,3.10674164626140];
d_32_34_5 = 1.000000000000000e-03;
lamb_32_34_5 = 0.001942858057143;
[AF_32_34_5] = Array_Fact(fliplr(phi_32_34_5),lamb_32_34_5,d_32_34_5);

phi_32_35 =[-0.400401765961208,1.79871327752466,-2.28535713148212,-
0.0862419357997771,2.11287268229751,-
1.97119891560109,0.227915529707824,2.42703073494784,-

```
1.65703958662065,0.542075410090806,2.74119088762857,-
1.34287937606514,0.856235522107660,3.05535071779001,-
1.02871997129229,1.17039373798869,-2.91367712388198,-
0.714561918641969,1.48455306696913,-2.59951724349900,-
0.400401765961228,1.79871327752464,-2.28535713148214,-0.0862419357997973,-
1.97119891560111,2.11287268229749,0.227915529707803,2.42703073494782,-
1.65703958662067,0.542075410090786,2.74119088762855,-1.34287937606516];
d_32_35 = 1.000000000000000e-03;
lamb_32_35 =0.00193548478178368;
[AF_32_35] = Array_Fact(fliplr(phi_32_35),lamb_32_35,d_32_35);

[Peak_32_25,I_32_25] = max(AF_32_25);
Angle_32_25=theta(I_32_25);
[Peak_32_25_5,I_32_25_5] = max(AF_32_25_5);
Angle_32_25_5 = theta(I_32_25_5);
[Peak_32_26,I_32_26] = max(AF_32_26);
Angle_32_26=theta(I_32_26);
[Peak_32_26_5,I_32_26_5] = max(AF_32_26_5);
Angle_32_26_5 = theta(I_32_26_5);
[Peak_32_27,I_32_27] = max(AF_32_27);
Angle_32_27=theta(I_32_27);
[Peak_32_27_5,I_32_27_5] = max(AF_32_27_5);
Angle_32_27_5=theta(I_32_27_5);
```

```
[Peak_32_28,I_32_28] = max(AF_32_28);
Angle_32_28=theta(I_32_28);
[Peak_32_28_5,I_32_28_5] = max(AF_32_28_5);
Angle_32_28_5=theta(I_32_28_5);
[Peak_32_29,I_32_29] = max(AF_32_29);
Angle_32_29=theta(I_32_29);
[Peak_32_29_5,I_32_29_5] = max(AF_32_29_5);
Angle_32_29_5=theta(I_32_29_5);
[Peak_32_30,I_32_30] = max(AF_32_30);
Angle_32_30=theta(I_32_30);
[Peak_32_30_5,I_32_30_5] = max(AF_32_30_5);
Angle_32_30_5=theta(I_32_30_5);
[Peak_32_31,I_32_31] = max(AF_32_31);
Angle_32_31=theta(I_32_31);
[Peak_32_31_5,I_32_31_5] = max(AF_32_31_5);
Angle_32_31_5=theta(I_32_31_5);
[Peak_32_32,I_32_32] = max(AF_32_32);
Angle_32_32=theta(I_32_32);
[Peak_32_32_5,I_32_32_5] = max(AF_32_32_5);
Angle_32_32_5=theta(I_32_32_5);
[Peak_32_33,I_32_33] = max(AF_32_33);
Angle_32_33 = theta(I_32_33);
[Peak_32_33_5,I_32_33_5] = max(AF_32_33_5);
```

```
Angle_32_33_5=theta(I_32_33_5);
[Peak_32_34,I_32_34] = max(AF_32_34);
Angle_32_34=theta(I_32_34);
[Peak_32_34_5,I_32_34_5] = max(AF_32_34_5);
Angle_32_34_5=theta(I_32_34_5);
[Peak_32_35,I_32_35] = max(AF_32_35);
Angle_32_35=theta(I_32_35);

Angles_32 =
[Angle_32_25,Angle_32_25_5,Angle_32_26,Angle_32_26_5,Angle_32_27,Angle_32_27_5,An
gle_32_28,Angle_32_28_5,Angle_32_29,Angle_32_29_5,Angle_32_30,Angle_32_30_5,Angle_
32_31,Angle_32_31_5,Angle_32_32,Angle_32_32_5,Angle_32_33,Angle_32_33_5,Angle_32_
34,Angle_32_34_5,Angle_32_35];

%2D Plot
figure(8)
plot(theta/pi*180,10*log10(AF_32_25))
hold on
plot(theta/pi*180,10*log10(AF_32_25_5))
plot(theta/pi*180,10*log10(AF_32_26))
plot(theta/pi*180,10*log10(AF_32_26_5))
plot(theta/pi*180,10*log10(AF_32_27))
plot(theta/pi*180,10*log10(AF_32_27_5))
```

```
plot(theta/pi*180,10*log10(AF_32_28))
plot(theta/pi*180,10*log10(AF_32_28_5))
plot(theta/pi*180,10*log10(AF_32_29))
plot(theta/pi*180,10*log10(AF_32_29_5))
plot(theta/pi*180,10*log10(AF_32_30))
plot(theta/pi*180,10*log10(AF_32_30_5))
plot(theta/pi*180,10*log10(AF_32_31))
plot(theta/pi*180,10*log10(AF_32_31_5))
plot(theta/pi*180,10*log10(AF_32_32))
plot(theta/pi*180,10*log10(AF_32_32_5))
plot(theta/pi*180,10*log10(AF_32_33))
plot(theta/pi*180,10*log10(AF_32_33_5))
plot(theta/pi*180,10*log10(AF_32_34))
plot(theta/pi*180,10*log10(AF_32_34_5))
plot(theta/pi*180,10*log10(AF_32_35))
title("32 Element Array Factor: Shift from 25GHz to 35GHz")
ylabel("Array Factor (dB)")
xlabel("Angle (°)")
hold off

figure(17)
polarplot(theta,AF_32_25)
hold on
```

```
polarplot(theta,AF_32_25_5)
polarplot(theta,AF_32_26)
polarplot(theta,AF_32_26_5)
polarplot(theta,AF_32_27)
polarplot(theta,AF_32_27_5)
polarplot(theta,AF_32_28)
polarplot(theta,AF_32_28_5)
polarplot(theta,AF_32_29)
polarplot(theta,AF_32_29_5)
polarplot(theta,AF_32_30)
polarplot(theta,AF_32_30_5)
polarplot(theta,AF_32_31)
polarplot(theta,AF_32_31_5)
polarplot(theta,AF_32_32)
polarplot(theta,AF_32_32_5)
polarplot(theta,AF_32_33)
polarplot(theta,AF_32_33_5)
polarplot(theta,AF_32_34)
polarplot(theta,AF_32_34_5)
polarplot(theta,AF_32_35)
title("32 Element Array Factor: Shift from 25GHz to 35GHz")
hold off
```

```
AF_32 = [10*log10(AF_32_25);
10*log10(AF_32_25_5);10*log10(AF_32_26);10*log10(AF_32_26_5);10*log10(AF_32_27);10
*log10(AF_32_27_5);10*log10(AF_32_28);
10*log10(AF_32_28_5);10*log10(AF_32_29);10*log10(AF_32_29_5);10*log10(AF_32_30);10
*log10(AF_32_30_5);10*log10(AF_32_31);
10*log10(AF_32_31_5);10*log10(AF_32_32);10*log10(AF_32_32_5);10*log10(AF_32_33);10
*log10(AF_32_33_5);10*log10(AF_32_34);10*log10(AF_32_34_5);10*log10(AF_32_35)];

%AF_32 = [AF_32_25;
AF_32_25_5;AF_32_26;AF_32_26_5;AF_32_27;AF_32_27_5;AF_32_28;
AF_32_28_5;AF_32_29;AF_32_29_5;AF_32_30;AF_32_30_5;AF_32_31;
AF_32_31_5;AF_32_32;AF_32_32_5;AF_32_33;AF_32_33_5;AF_32_34;AF_32_34_5;AF_32
_35];
figure(15)
surf(theta,F,AF_32)
hold on
title('32 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off

figure(74)
```

```
mesh(theta/pi*180,F,AF_32)
hold on
title('32 Element Array Factor: Shift from 25GHz to 35GHz')
xlabel('Angle (°)')
ylabel('Frequency (GHz)')
zlabel('Array Factor (dB)')
hold off
```

Steering_Angle.m

```
tau_IF = 7*(10^-11);

tau_LO = 1.75*(10^-11);

f_IF= linspace(25,35,100);

f_IF_10= linspace(25,35,11);

angles2_0=asin(0.14*(f_IF-30))/pi*180;

theta = linspace(-1,1);

f_IF_21 = linspace(25,35,21);

w_0 = 2*pi*30*(10^9);

w_IF = 2*pi*f_IF*(10^9);

w_LO =2*pi*120*(10^9);

part1 = ((-1*tau_IF.*(w_0-w_IF))./(pi));

part2 =((tau_IF.*w_LO)./((tau_IF.*(w_0-w_IF))+(w_LO*(tau_IF+tau_LO))));

steering_angle_theory = asin(part1.*part2);

steering = asin(((2^0.5)/2)*theta);


figure(21)

%plot(f_IF,(steering_angle_theory.*180)./(pi))

plot(f_IF_21,((Angles_4.*180)./(pi)),'*', 'markersize', 10)

hold on

plot(f_IF,angles2_0')

plot(f_IF,8*(f_IF-30),'--')

title("4 Element Steering Angle: Simulation vs Theory")
```

```
ylabel("Angle (°)")
xlabel("Frequency (GHz)")
legend("Simulation","Theory","Derivative at Center Frequency")
hold off

figure(22)
plot(f_IF_21,((Angles_8.*180)./(pi)),'*', 'markersize', 10)
hold on
plot(f_IF,angles2_0')
plot(f_IF,8*(f_IF-30),'--')
title("8 Element Steering Angle: Simulation vs Theory")
ylabel("Angle (°)")
xlabel("Frequency (GHz)")
legend("Simulation","Theory","Derivative at Center Frequency")
hold off

figure(23)
plot(f_IF_21,((Angles_16.*180)./(pi)),'*', 'markersize', 10)
hold on
plot(f_IF,angles2_0')
plot(f_IF,8*(f_IF-30),'--')
title("16 Element Steering Angle: Simulation vs Theory")
ylabel("Angle (°)")
```

```
xlabel("Frequency (GHz)")
legend("Simulation","Theory","Derivative at Center Frequency")
hold off

figure(24)
plot(f_IF_21,((Angles_32.*180)./(pi)),'*', 'markersize', 10)
hold on
plot(f_IF,angles2_0')
plot(f_IF,8*(f_IF-30),'--')
title("32 Element Steering Angle: Simulation vs Theory")
ylabel("Angle (°)")
xlabel("Frequency (GHz)")
legend("Simulation","Theory","Derivative at Center Frequency")
hold off
```

Beamwidth.m

```
function [Width] = Beamwidth(AF)
   theta =linspace(-pi/2,pi/2,10001);
   M = max(AF);
   M
   I = find(abs(AF-M+3)< 0.1);
   theta(I)
   Width = abs(theta(I(1))).*2;
End
```

Array_Factor.m

```
function [A1] = Array_Fact(phi,lamda,d)
   theta = linspace(-pi/2,pi/2,10001);
   if (length(phi) == 4)
      phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
      phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
      phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
      phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
      %AF = (abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4))).^2;
      new1 = chebwin(4,20);
      A1 = (abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4))).^2;
%       A2 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
%       new1 = chebwin(4,40);
%       A3 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
%       new1 = chebwin(4,50);
%       A4 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
%
```

```
%
    elseif (length(phi) == 8)
        phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
        phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
        phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
        phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
        phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
        phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
        phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
        phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
        %AF =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8))).^2;
        new1 = chebwin(8,18);
        A1 =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8))).^2;
%       A2 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*ex
p(1i*phi_8))).^2;
%       new1 = chebwin(8,22);
```

```
%       A3 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*ex
p(1i*phi_8))).^2;
%       new1 = chebwin(8,30);
%       A4 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*ex
p(1i*phi_8))).^2;
%
   elseif (length(phi) == 16)
       phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
       phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
       phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
       phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
       phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
       phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
       phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
       phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
       phi_9 = phi(9)+16*pi/lamda*sin(theta)*d;
       phi_10 = phi(10)+18*pi/lamda*sin(theta)*d;
       phi_11 = phi(11)+20*pi/lamda*sin(theta)*d;
       phi_12 = phi(12)+22*pi/lamda*sin(theta)*d;
```

```
phi_13 = phi(13)+24*pi/lamda*sin(theta)*d;
phi_14 = phi(14)+26*pi/lamda*sin(theta)*d;
phi_15 = phi(15)+28*pi/lamda*sin(theta)*d;
phi_16 = phi(16)+30*pi/lamda*sin(theta)*d;
%AF =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)
+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16))).^2;
new = chebwin(16,18);
A1 =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)
+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16))).^2;
%       A2 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;
%       new = chebwin(16,22);
%       A3 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
```

```
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;
%       new = chebwin(16,30);
%       A4 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;
%
   elseif (length(phi) == 32)
       phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
       phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
       phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
       phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
       phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
       phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
       phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
       phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
       phi_9 = phi(9)+16*pi/lamda*sin(theta)*d;
       phi_10 = phi(10)+18*pi/lamda*sin(theta)*d;
       phi_11 = phi(11)+20*pi/lamda*sin(theta)*d;
```

```
phi_12 = phi(12)+22*pi/lamda*sin(theta)*d;

phi_13 = phi(13)+24*pi/lamda*sin(theta)*d;

phi_14 = phi(14)+26*pi/lamda*sin(theta)*d;

phi_15 = phi(15)+28*pi/lamda*sin(theta)*d;

phi_16 = phi(16)+30*pi/lamda*sin(theta)*d;

phi_17 = phi(17)+32*pi/lamda*sin(theta)*d;

phi_18 = phi(18)+34*pi/lamda*sin(theta)*d;

phi_19 = phi(19)+36*pi/lamda*sin(theta)*d;

phi_20= phi(20)+38*pi/lamda*sin(theta)*d;

phi_21 = phi(21)+40*pi/lamda*sin(theta)*d;

phi_22 = phi(22)+42*pi/lamda*sin(theta)*d;

phi_23 = phi(23)+44*pi/lamda*sin(theta)*d;

phi_24 = phi(24)+46*pi/lamda*sin(theta)*d;

phi_25 = phi(25)+48*pi/lamda*sin(theta)*d;

phi_26 = phi(26)+50*pi/lamda*sin(theta)*d;

phi_27 = phi(27)+52*pi/lamda*sin(theta)*d;

phi_28 = phi(28)+54*pi/lamda*sin(theta)*d;

phi_29 = phi(29)+56*pi/lamda*sin(theta)*d;

phi_30 = phi(30)+58*pi/lamda*sin(theta)*d;

phi_31 = phi(31)+60*pi/lamda*sin(theta)*d;

phi_32 = phi(32)+62*pi/lamda*sin(theta)*d;

%AF

=(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_
```

```
6)+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_1
2)+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16)+exp(1i*phi_17)+exp(1i*p
hi_18)+exp(1i*phi_19)+exp(1i*phi_20)+exp(1i*phi_21)+exp(1i*phi_22)+exp(1i*phi_23)+exp(
1i*phi_24)+exp(1i*phi_25)+exp(1i*phi_26)+exp(1i*phi_27)+exp(1i*phi_28)+exp(1i*phi_29)+
exp(1i*phi_30)+exp(1i*phi_31)+exp(1i*phi_32))).^2;
        new = chebwin(32,18);
        A1 =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)
+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16)+exp(1i*phi_17)+exp(1i*phi
_18)+exp(1i*phi_19)+exp(1i*phi_20)+exp(1i*phi_21)+exp(1i*phi_22)+exp(1i*phi_23)+exp(1i
*phi_24)+exp(1i*phi_25)+exp(1i*phi_26)+exp(1i*phi_27)+exp(1i*phi_28)+exp(1i*phi_29)+ex
p(1i*phi_30)+exp(1i*phi_31)+exp(1i*phi_32))).^2;
%       A2 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))
+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*e
xp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_2
5))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29
```

)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_3
2))))).^2;

% new = chebwin(32,22);

% A3 =

(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))
+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*e
xp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_2
5))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29
)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_3
2))))).^2;

% new = chebwin(32,30);

% A4 =

(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))
+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*e
xp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_2

```
5))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29
)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_3
2))))).^2;


    else
        AF ="NOT WORKING"
    end


end
```

Array_Factor_4.m

```
function [A1,A2,A3,A4] = Array_Fact_4(phi,lamda,d)
    theta = linspace(-pi/2,pi/2,10001);
    if (length(phi) == 4)
        phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
        phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
        phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
        phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
        %AF = (abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4))).^2;
        new1 = chebwin(4,20);
        A1 = (abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4))).^2;
        A2 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
        new1 = chebwin(4,40);
        A3 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
        new1 = chebwin(4,50);
        A4 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4))).^2;
```

```
    elseif (length(phi) == 8)
        phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
        phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
        phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
        phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
        phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
        phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
        phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
        phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
        %AF =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8))).^2;
        new1 = chebwin(8,18);
        A1 =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8))).^2;
        A2 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i
*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*ex
p(1i*phi_8))).^2;
        new1 = chebwin(8,22);
```

```
        A3 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*exp(1i*phi_8))).^2;
        new1 = chebwin(8,30);
        A4 =
(abs(new1(1)*exp(1i*phi_1)+new1(2)*exp(1i*phi_2)+new1(3)*exp(1i*phi_3)+new1(4)*exp(1i*phi_4)+new1(5)*exp(1i*phi_5)+new1(6)*exp(1i*phi_6)+new1(7)*exp(1i*phi_7)+new1(8)*exp(1i*phi_8))).^2;

    elseif (length(phi) == 16)
        phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
        phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
        phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
        phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
        phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
        phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
        phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
        phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
        phi_9 = phi(9)+16*pi/lamda*sin(theta)*d;
        phi_10 = phi(10)+18*pi/lamda*sin(theta)*d;
        phi_11 = phi(11)+20*pi/lamda*sin(theta)*d;
        phi_12 = phi(12)+22*pi/lamda*sin(theta)*d;
```

```
phi_13 = phi(13)+24*pi/lamda*sin(theta)*d;
phi_14 = phi(14)+26*pi/lamda*sin(theta)*d;
phi_15 = phi(15)+28*pi/lamda*sin(theta)*d;
phi_16 = phi(16)+30*pi/lamda*sin(theta)*d;
%AF =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)
+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16))).^2;
new = chebwin(16,18);
A1 =
(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)
+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)
+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16))).^2;
A2 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;
new = chebwin(16,22);
A3 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
```

```
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;
        new = chebwin(16,30);
        A4 =
(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i
*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*e
xp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+
(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*ex
p(1i*phi_15))+(new(16)*exp(1i*phi_16)))).^2;

    elseif (length(phi) == 32)
        phi_1 = phi(1)+0*pi/lamda*sin(theta)*d;
        phi_2 = phi(2)+2*pi/lamda*sin(theta)*d;
        phi_3 = phi(3)+4*pi/lamda*sin(theta)*d;
        phi_4 = phi(4)+6*pi/lamda*sin(theta)*d;
        phi_5 = phi(5)+8*pi/lamda*sin(theta)*d;
        phi_6 = phi(6)+10*pi/lamda*sin(theta)*d;
        phi_7 = phi(7)+12*pi/lamda*sin(theta)*d;
        phi_8 = phi(8)+14*pi/lamda*sin(theta)*d;
        phi_9 = phi(9)+16*pi/lamda*sin(theta)*d;
        phi_10 = phi(10)+18*pi/lamda*sin(theta)*d;
        phi_11 = phi(11)+20*pi/lamda*sin(theta)*d;
```

```
phi_12 = phi(12)+22*pi/lamda*sin(theta)*d;

phi_13 = phi(13)+24*pi/lamda*sin(theta)*d;

phi_14 = phi(14)+26*pi/lamda*sin(theta)*d;

phi_15 = phi(15)+28*pi/lamda*sin(theta)*d;

phi_16 = phi(16)+30*pi/lamda*sin(theta)*d;

phi_17 = phi(17)+32*pi/lamda*sin(theta)*d;

phi_18 = phi(18)+34*pi/lamda*sin(theta)*d;

phi_19 = phi(19)+36*pi/lamda*sin(theta)*d;

phi_20= phi(20)+38*pi/lamda*sin(theta)*d;

phi_21 = phi(21)+40*pi/lamda*sin(theta)*d;

phi_22 = phi(22)+42*pi/lamda*sin(theta)*d;

phi_23 = phi(23)+44*pi/lamda*sin(theta)*d;

phi_24 = phi(24)+46*pi/lamda*sin(theta)*d;

phi_25 = phi(25)+48*pi/lamda*sin(theta)*d;

phi_26 = phi(26)+50*pi/lamda*sin(theta)*d;

phi_27 = phi(27)+52*pi/lamda*sin(theta)*d;

phi_28 = phi(28)+54*pi/lamda*sin(theta)*d;

phi_29 = phi(29)+56*pi/lamda*sin(theta)*d;

phi_30 = phi(30)+58*pi/lamda*sin(theta)*d;

phi_31 = phi(31)+60*pi/lamda*sin(theta)*d;

phi_32 = phi(32)+62*pi/lamda*sin(theta)*d;

%AF

=(abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_
```

6)+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16)+exp(1i*phi_17)+exp(1i*phi_18)+exp(1i*phi_19)+exp(1i*phi_20)+exp(1i*phi_21)+exp(1i*phi_22)+exp(1i*phi_23)+exp(1i*phi_24)+exp(1i*phi_25)+exp(1i*phi_26)+exp(1i*phi_27)+exp(1i*phi_28)+exp(1i*phi_29)+exp(1i*phi_30)+exp(1i*phi_31)+exp(1i*phi_32))).^2;

new = chebwin(32,18);

A1 = (abs(exp(1i*phi_1)+exp(1i*phi_2)+exp(1i*phi_3)+exp(1i*phi_4)+exp(1i*phi_5)+exp(1i*phi_6)+exp(1i*phi_7)+exp(1i*phi_8)+exp(1i*phi_9)+exp(1i*phi_10)+exp(1i*phi_11)+exp(1i*phi_12)+exp(1i*phi_13)+exp(1i*phi_14)+exp(1i*phi_15)+exp(1i*phi_16)+exp(1i*phi_17)+exp(1i*phi_18)+exp(1i*phi_19)+exp(1i*phi_20)+exp(1i*phi_21)+exp(1i*phi_22)+exp(1i*phi_23)+exp(1i*phi_24)+exp(1i*phi_25)+exp(1i*phi_26)+exp(1i*phi_27)+exp(1i*phi_28)+exp(1i*phi_29)+exp(1i*phi_30)+exp(1i*phi_31)+exp(1i*phi_32))).^2;

A2 = (abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*exp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*exp(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*exp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_25))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29

)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_32))))).^2;

new = chebwin(32,22);

A3 =

(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*exp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*exp(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*exp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_25))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_32))))).^2;

new = chebwin(32,30);

A4 =

(abs((new(1)*exp(1i*phi_1))+(new(2)*exp(1i*phi_2))+(new(3)*exp(1i*phi_3))+(new(4)*exp(1i*phi_4))+(new(5)*exp(1i*phi_5))+(new(6)*exp(1i*phi_6))+(new(7)*exp(1i*phi_7))+(new(8)*exp(1i*phi_8))+(new(9)*exp(1i*phi_9))+(new(10)*exp(1i*phi_10))+(new(11)*exp(1i*phi_11))+(new(12)*exp(1i*phi_12))+(new(13)*exp(1i*phi_13))+(new(14)*exp(1i*phi_14))+(new(15)*exp(1i*phi_15))+(new(16)*exp(1i*phi_16)+(new(17)*exp(1i*phi_17))+(new(18)*exp(1i*phi_18))+(new(19)*exp(1i*phi_19))+(new(20)*exp(1i*phi_20))+(new(21)*exp(1i*phi_21))+(new(22)*exp(1i*phi_22))+(new(23)*exp(1i*phi_23))+(new(24)*exp(1i*phi_24))+(new(25)*exp(1i*phi_2

```
5))+(new(26)*exp(1i*phi_26))+(new(27)*exp(1i*phi_27))+(new(28)*exp(1i*phi_28))+(new(29
)*exp(1i*phi_29))+(new(30)*exp(1i*phi_30))+(new(31)*exp(1i*(31)))+(new(32)*exp(1i*phi_3
2))))).^2;


    else
        AF ="NOT WORKING"
    end


end
```

Beamwidth_vs_SLL.m

```
% 4 Element

%SLL4_20 =

Beam4_20 =18.612000000000002;

%SLL4_25 =

Beam4_25 = 19.512000000000000;

%SLL4_30 =

Beam4_30 =20.160000000000000;

%SLL4_35 =

Beam4_35 = 20.592000000000000;

%SLL4_40 =

Beam4_40 = 20.916000000000000;

%SLL4_45 =

Beam4_45 = 21.132000000000000;

%SLL4_50 =

Beam4_50 = 21.276000000000000;

%SLL4_55 =

Beam4_55 = 21.384000000000000;

%SLL4_60 =

Beam4_60 = 21.456000000000000;
```

```
Beam4
=[Beam4_20,Beam4_25,Beam4_30,Beam4_35,Beam4_40,Beam4_45,Beam4_50,Beam4_55,Be
am4_60];

n =4
lambda = 0.002;
d = 1.000000000000000e-03;
L = n*d;

theta_0 = pi/2;
SLL = linspace(20,60);
SLL_9= linspace(20,60,9);

Ro = 10.^(SLL/20)

theta_h = acos((cos(theta_0))-(0.443.*((lambda)/(L+d)))) -
acos((cos(theta_0))+(0.443.*((lambda)./(L+d))))
f = 1 + 0.636.*(((2./Ro).*cosh(((acosh(Ro).^2)-(pi.^2)).^0.5)).^2)
Directivity = (2.*(Ro.^2))./(1+(((Ro.^2) - 1).*f.*((lambda)./(L))))
tape_E =Directivity./n;
Beamwidth = theta_h*f*(180/pi)
figure(30)
plot(SLL,Beamwidth);
```

```
hold on

plot(SLL_9,Beam4,'*');

title("4 Element Beamwidth")

xlabel("Side Lobe Level(dB)")

ylabel("Angle (°)")

legend("Theory","Data")

hold off


% 8 Element

%SLL8_20 =

Beam8_20 = 10.224000000000002;

%SLL8_25 =

Beam8_25 = 11.088000000000001;

%SLL8_30 =

Beam8_30 = 11.880000000000000;

%SLL8_35 =

Beam8_35 = 12.527999999999999;

%SLL8_40 =

Beam8_40 = 13.104000000000000;

%SLL8_45 =

Beam8_45 = 13.607999999999999;

%SLL8_50 =

Beam8_50 = 14.040000000000000;
```

```
%SLL8_55 =

Beam8_55 = 14.400000000000002;

%SLL8_60 =

Beam8_60 = 14.724000000000000;

Beam8

=[Beam8_20,Beam8_25,Beam8_30,Beam8_35,Beam8_40,Beam8_45,Beam8_50,Beam8_55,Be

am8_60];

n =8

lambda = 0.002;

d = 1.000000000000000e-03;

L = n*d;


theta_0 = pi/2;

SLL = linspace(20,60);

SLL_9= linspace(20,60,9);


Ro = 10.^(SLL/20)


theta_h = acos((cos(theta_0))-(0.443.*((lambda)/(L+d)))) -

acos((cos(theta_0))+(0.443.*((lambda)./(L+d))))

f = 1 + 0.636.*(((2./Ro).*cosh(((acosh(Ro).^2)-(pi.^2)).^0.5)).^2)

Directivity = (2.*(Ro.^2))./(1+(((Ro.^2) - 1).*f.*((lambda)./(L))))

tape_E =Directivity./n;
```

```
Beamwidth = theta_h*f*(180/pi)
figure(31)
plot(SLL,Beamwidth);
hold on
plot(SLL_9,Beam8,'*');
title("8 Element Beamwidth")
xlabel("Side Lobe Level(dB)")
ylabel("Angle (°)")
legend("Theory","Data")
hold off

% 32 Element
%SLL32_20 =
Beam32_20 = 2.340000000000000;
%SLL32_25 =
Beam32_25 = 2.556000000000001;
%SLL32_30 =
Beam32_30 = 2.772000000000000;
%SLL32_35 =
Beam32_35 = 2.952000000000000;
%SLL32_40 =
Beam32_40 = 3.132000000000000;
%SLL32_45 =
```

```
Beam32_45 = 3.312000000000000;

%SLL32_50 =

Beam32_50 = 3.492000000000000;

%SLL32_55 =

Beam32_55 = 3.636000000000000;

%SLL32_60 =

Beam32_60 = 3.780000000000000;

Beam32=[Beam32_20,Beam32_25,Beam32_30,Beam32_35,Beam32_40,Beam32_45,Beam32_

50,Beam32_55,Beam32_60];

n =32

lambda = 0.002;

d = 1.000000000000000e-03;

L = n*d;


theta_0 = pi/2;

SLL = linspace(20,60);

SLL_9= linspace(20,60,9);


Ro = 10.^(SLL/20)


theta_h = acos((cos(theta_0))-(0.443.*((lambda)/(L+d)))) -

acos((cos(theta_0))+(0.443.*((lambda)./(L+d))))

f = 1 + 0.636.*(((2./Ro).*cosh(((acosh(Ro).^2)-(pi.^2)).^0.5)).^2)
```

```
Directivity = (2.*(Ro.^2))./(1+(((Ro.^2) - 1).*f.*((lambda)./(L))))

tape_E =Directivity./n;

Beamwidth = theta_h*f*(180/pi)

figure(32)

plot(SLL,Beamwidth);

hold on

plot(SLL_9,Beam32,'*');

title("32 Element Beamwidth")

xlabel("Side Lobe Level(dB)")

ylabel("Angle (°)")

legend("Theory","Data")

hold off


% 16 Element

%SLL16_20 =

Beam16_20 = 4.932000000000000;

%SLL16_20_5 =

Beam16_25 = 5.364000000000000;

%SLL16_30 =

Beam16_30 = 5.760000000000000;

%SLL16_30_5 =

Beam16_35 = 6.120000000000000;

%SLL16_40 =
```

```
Beam16_40 = 6.480000000000000;

%SLL16_40_5 =

Beam16_45 = 6.803999999999999;

%SLL16_50 =

Beam16_50 = 7.092000000000000;

%SLL16_50_5 =

Beam16_55 =7.380000000000000;

%SLL16_60 =

Beam16_60 =7.632000000000001;

Beam16=[Beam16_20,Beam16_25,Beam16_30,Beam16_35,Beam16_40,Beam16_45,Beam16_

50,Beam16_55,Beam16_60];

n =16

lambda = 0.002;

d = 1.000000000000000e-03;

L = n*d;


theta_0 = pi/2;

SLL = linspace(20,60);

SLL_9= linspace(20,60,9);


Ro = 10.^(SLL/20)
```

```
theta_h = acos((cos(theta_0))-(0.443.*((lambda)/(L+d)))) -
acos((cos(theta_0))+(0.443.*((lambda)./(L+d))))
f = 1 + 0.636.*(((2./Ro).*cosh(((acosh(Ro).^2)-(pi.^2)).^0.5)).^2)
Directivity = (2.*(Ro.^2))./(1+(((Ro.^2) - 1).*f.*((lambda)./(L))))
tape_E =Directivity./n;
Beamwidth = theta_h*f*(180/pi)
figure(33)
plot(SLL,Beamwidth);
hold on
plot(SLL_9,Beam16,'*');
title("16 Element Beamwidth")
xlabel("Side Lobe Level(dB)")
ylabel("Angle (°)")
legend("Theory","Data")
hold off
```

BIBLIOGRAPHY

[1] Schneider, David & Rosch, Markus & Tessmann, Axel & Zwick, Thomas. (2019). A Low-Loss W-Band Frequency-Scanning Antenna for Wideband Multichannel Radar Applications. IEEE Antennas and Wireless Propagation Letters. PP. 1-1. 10.1109/LAWP.2019.2904170.

[2] W. Mayer, M. Wetzel and W. Menzel, "A novel direct-imaging radar sensor with frequency scanned antenna," IEEE MTT-S International Microwave Symposium Digest, 2003, Philadelphia, PA, USA, 2003, pp. 1941-1944 vol.3, doi: 10.1109/MWSYM.2003.1210538

[3] H. V. Nguyen, S. Abielmona, A. Rennings and C. Caloz, "Pencil-Beam Full-Space Scanning 2D CRLH Leaky-Wave Antenna Array," 2007 International Symposium on Signals, Systems and Electronics, Montreal, QC, Canada, 2007, pp. 139-142, doi: 10.1109/ISSSE.2007.4294433.

[4] P. Lu et al., "InP-Based THz Beam Steering Leaky-Wave Antenna," in IEEE Transactions on Terahertz Science and Technology, vol. 11, no. 2, pp. 218-230, March 2021, doi: 10.1109/TTHZ.2020.3039460.

[5] A. Lai, T. Itoh and C. Caloz, "Composite right/left-handed transmission line metamaterials," in IEEE Microwave Magazine, vol. 5, no. 3, pp. 34-50, Sept. 2004, doi: 10.1109/MMW.2004.1337766.

[6] T. Geibig, A. Shoykhetbrod, A. Hommes, R. Herschel and N. Pohl, "Compact 3D imaging radar based on FMCW driven frequency-scanning antennas," 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 2016, pp. 1-5, doi: 10.1109/RADAR.2016.7485168.

[7] A. Taslimi, W. Alomar and A. Mortazawi, "Phase compensated serially fed array using the antenna as a part of negative group delay circuit," 2015 IEEE MTT-S International Microwave Symposium, Phoenix, AZ, USA, 2015, pp. 1-4, doi: 10.1109/MWSYM.2015.7166902.

[8] J. H. Choi, J. S. Sun and T. Itoh, "Frequency-Scanning Phased-Array Feed Network Based on Composite Right/Left-Handed Transmission Lines," in IEEE Transactions on Microwave Theory and Techniques, vol. 61, no. 8, pp. 3148-3157, Aug. 2013, doi: 10.1109/TMTT.2013.2263508.

[9] D. Ma et al., "Single-Shot Frequency-Diverse Near-Field Imaging Using High-Scanning-Rate Leaky-Wave Antenna," in IEEE Transactions on Microwave Theory and Techniques, doi: 10.1109/TMTT.2021.3065713.

[10] Ghasempour, Y., Shrestha, R., Charous, A. et al. Single-shot link discovery for terahertz wireless networks. Nat Commun 11, 2017 (2020). https://doi.org/10.1038/s41467-020-15761-4.

[11] M. H. Rahmani and D. Deslandes, "Backward to Forward Scanning Periodic Leaky-Wave Antenna With Wide Scanning Range," in IEEE Transactions on Antennas and Propagation, vol. 65, no. 7, pp. 3326-3335, July 2017, doi: 10.1109/TAP.2017.2705021.

[12] Robert Mailloux, Phased Array Antenna Handbook, Third Edition , Artech, 2017.

ACADEMIC VITA

PROFESSIONAL EXPERIENCE

NAVAL SURFACE WARFARE CENTER **University Park, PA**
Researcher
Jun 2021 – Present

- Joined Navy funded research group into
- Worked under the instruction of Dr. Yan Li
- Used applications such as Mininet and Wireshark to determine network latency

SMART RESPONSIVE FACADES PROJECT **University Park, PA**
Part-Time Researcher
Jun 2021 – Present

- Designed a circuit for actuating the façade in response to the light and temperature
- Assisted in creating the structure of the façade device

PROJECT DRAWDOWN
University Park, PA *Research Internship*
Jun 2021 – Aug 2021

- Joined NSF funded research project to develop climate change solutions
- Worked with Dr. Urbina and researchers from Peru to help with existing solar filtration project
- Summarized findings in an online symposium

WIND ENERGY RESEARCH **University Park, PA**
Research Internship
May 2020 – Aug 2020

- Joined prestigious NSF funded research project
- Developed robust wind farm control systems in MATLAB and MATLAB Simulink
- Produced a comprehensive final report summarizing our findings

EDUCATION

THE PENNSYLVANIA STATE UNIVERSITY **University Park, PA**
Electrical Engineering, May 2022

- Schreyer Honors College Member

Selected Courses:

RF and mmWave Integrated Circuits
RF and Microwave Engineering
Energy Conversion
Linear Control Systems
Fundamental Discrete Signal Processing

ACTIVITIES

ETA KAPPA NU - EPSILON CHAPTER **University Park, PA**
Secretary/Corporate Liaison
Jan 2021 – Present

- Gained membership through being top 25% of class
- Organized corporate sponsored events and workshops
- Kept close contact with corporate affiliates

ALPHA DELTA PHI FRATERNITY **University Park, PA**
Treasurer/Executive Board Member
Dec 2019 – Jan 2021

- Managed and organized the fraternity finances with dues as well as utilities
- Developed ways to reduce cost and generate more income
- Coordinated with community to build outreach and philanthropic events

TECHNICAL SKILLS

- Programming in Python and C++
- Constructing simulations and MATLAB and MATLAB Simulink
- Building robust control systems in LabVIEW