



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Techniques for suppressing grating lobes

W. Deyzel

21750793

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Mr L. Grootboom

October 2021

Acknowledgements

I would like to thank my family for providing me with the opportunity to pursue my dreams, my friends for their support during my engineering degree and my Supervisor for guiding me in the right direction for my thesis.



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / Plagiarism Declaration

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

Plagiarism is the use of ideas, material and other intellectual property of another's work and to present it as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.

I also understand that direct translations are plagiarism.

4. Dienooreenkomsdig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / Student number	Handtekening / Signature
Voorletters en van / Initials and surname	Datum / Date

Abstract

English

Antennas radiating uniformly in all directions are known as omnidirectional antennas and can be placed in a linear array (straight line) to increase the overall antenna gain. Improving the antenna gain means data can be transmitted further. Two methods of beam steering are possible: namely mechanical steering and phased arrays. Mechanical steering physically moves the antenna in the target direction which is a slow process due to the sheer size of the antennas. Phased arrays supply a phase shift to each antenna's signal to digitally steer the beam to the desired target direction. This allows for instantaneous steering abilities through altering digital parameters but introduces grating lobes (wasted energy thus reducing the performance) when steered beyond a critical angle. This report investigates grating lobes and how they can be suppressed using the digital parameters and optimizing the placement of the antennas. The main factor that influences the position of grating lobes is the antenna spacings. A genetic algorithm is designed and deployed to find the optimal spacing of an antenna array of six elements and is also used to steer the beam to any target direction. The non-uniform spaced antenna array is then compared to a uniformly spaced antenna array of half a wavelength. It is shown that the non-uniform spaced antenna array has lower sidelobe levels and suppresses the grating lobes that appear in the uniformly spaced antenna array but have wider half-power bandwidth. This results in a more directive antenna and depending on the application wider steering range.

Afrikaans

If the English version is correct I will translate it to Afrikaans.

Contents

Declaration	ii
Abstract	iii
List of Figures	vi
List of Tables	vii
Nomenclature	viii
1. Introduction	1
1.1. Problem statement	2
1.2. Objective	2
1.3. Scope	3
2. Literature Review	4
2.1. Array Geometry	5
2.1.1. Uniformly Spaced Antenna Array	5
2.1.2. Symmetrical Non-Uniformly Spaced Antenna Array	5
2.2. Amplitude Distribution	6
2.3. Grating Lobes	6
2.4. Genetic Algorithm	6
2.5. Antenna Types	6
3. Design and Simulation	8
3.1. Array Factor	8
3.2. Fourier Transform and the AF	9
3.3. Genetic Algorithm	12
3.3.1. Fitness score	13
3.3.2. Selection	14
3.3.3. Genome	14
3.3.4. Crossover	14
3.3.5. Mutation	15
3.4. Chapter Summary	15

4. Results	16
4.1. Uniform linear Antenna Array Results.	17
4.1.1. Amplitude Distribution	17
4.1.2. Uniform linear Antenna Array Beam Steering	18
4.1.3. GA Optimizes Phase Shift for ULA.	20
4.2. Uniformly spaced array	22
4.3. Symmetrical non-uniformly spaced array	27
4.4. Chapter Summary	29
5. Conclusion	30
5.1. Summary	30
5.2. Future Work	31
Bibliography	32
A. Project Planning Schedule	33
B. Outcomes Compliance	34
C. More Results	36
C.0.1. Computer code (python) Genetic Algorithm for optimizing the different parameters of a LAA.	36

List of Figures

2.1. Antenna Types and Antenna Patterns [1]	7
3.1. Visual representation of the Fourier transform of a LAA.	10
3.4. Crossover process	15
3.5. Mutation process	15
4.1. Half-wave dipole antenna radiation pattern.	16
4.2. 3D radiation pattern for 6 element LAA.	16
4.3. Uniform vs Dolph-Chebyshev vs Binomial Amplitude vs GA amplitude distribution.	17
4.4. Beam steering (with equation (4.1)) with a uniform LAA of 6 antennas elements spaced 0.5λ apart.	18
4.5. The AF for Uniform LAA spaced 0.5λ apart steered using the GA.	20
4.6. The AF for a uniform LAA spaced from 0.5λ to 5λ apart.	22
4.7. Six antennas separated by a constant value $d = x\lambda$	23
4.8. Uniform LAA with 6 elements spaced 0.17λ apart.	24
4.9. Uniform LAA with 6 elements spaced 0.17λ apart top view.	24
4.10. Uniform LAA with 6 elements spaced 0.5λ apart with no grating lobes.	25
4.11. Uniform LAA with 6 elements spaced 0.5λ apart top view.	25
4.12. Uniform LAA with 6 elements spaced 2λ apart.	26
4.13. Uniform LAA with 6 elements spaced 2λ apart top view.	26
4.14. Steering beam from boresight to different target directions.	27
C.1. GA that optimizes the amplitude of the signal entering a LAA of size N to any desired direction.	36
C.2. GA that only optimizes the phase for a LAA of size N to any desired direction.	37
C.3. GA that optimizes the antenna element spacing and the phase shift for a LAA of size N to any desired direction.	38

List of Tables

4.1. Antenna Setup	16
4.2. The performance of a Uniform LAA spaced 0.5λ apart steered using equation (4.1).	18
4.3. The performance of a uniform LAA spaced 0.5λ apart steered using the GA.	21
4.4. Mathematical predictions vs AF calculation for the location of grating lobes.	23
4.5. Element Spacing that yield best grating lobe suppression.	28
4.6. Performance of non-uniformly spaced solutions.	28
B.1. Outcome and Compliance.	35

Nomenclature

Variables and functions

I	Current excitation of antenna element.
k	Wave number.
c_0	Speed of light in a vacuum.
λ	Wavelength of antenna signal.
f	Operating frequency of antenna.
$\Delta\delta$	Phase shift.
θ_T	Target Direction

Acronyms and abbreviations

LAA	Linear Antenna Array
ULA	Uniform Linear Array
ULAA	Uniform Linear Antenna Array
GA	Genetic Algorithm
SLL	Side-Lobe Level
GLL	Grating-Lobe Level
HPBW	Half Power Beamwidth
FT	Fourier Transform
GL	Grating Lobes
AF	Array Factor

Chapter 1

Introduction

Antennas are designed to radiate Electromagnetic energy. It is used for long-distance communication and/or transmitting of data [2]. In the simplest form, an antenna can be modelled as a point source radiating in all directions at a particular frequency. These point source antennas are known as omnidirectional antennas. The antennas can be arranged in a linear array (straight line) or a planar array (surface) with the distance between the antennas dependent on the application of the antenna array. Arranging antennae in an array improves the overall power gain of the net radiation pattern which improves its directivity. This is a result of the principle of superposition because each antenna contributes to the overall radiation pattern. It means that the radiation energy is more concentrated in the target direction. This result makes the transmission of data over a larger distance possible. To steer the beam of the antennas in the desired direction the antennas can be moved mechanically. The National Radiation Astronomy Observatory antenna array in Figure (1.1a) are 25 meters in diameter and weighs 230 tons [3], making mechanical steering difficult and slow. South Africa has a large radio telescope array called MeerKat Figure (1.1b) with a height of 19.5 meters and weighs 40 tons. Due to the sheer size of the antennas the moment of inertia is large which means more energy is needed to control the rotation of the antenna. This limits potential applications that require fast rotation of an antenna such as radar to track missiles.

A solution to the problem is to steer the radiating beam of the antenna array digitally through altering digital parameters such as the phase shift and amplitude distribution of the signal entering each antenna in the antenna array. This allows the radiating beam of the antenna array to be steered much faster, almost instantaneously, due to not having any moving parts. In Colorado USA, an antenna phased array see Figure (1.1c) is used to track multiple space missiles at the same time by steering the beam over the entire scanning range very quickly.

This method introduces grating lobes (GL) when the beam is steered beyond a critical angle. This means that an antenna array that uses non-mechanical beam steering has a limited range that the main beam can be steered to before GL's appear. These lobes are undesirable because it results in power loss decreasing the efficiency of the antenna array.



(a) National Radio Astronomy Observatory (NRAO) 80.47 km west of Socorro, New Mexico [3]



(b) South Africa's radio telescope array "MeerKAT" located in the Northern Cape. [4]



(c) Eglin air force base phased array radar for tracking space objects, [5]

GL also cause interference of transmitted or received data from undesirable angles.

1.1. Problem statement

In this project we will investigate how the parameters (eg. element spacing, phase shift and amplitude distributions of each element) affect grating lobes and how these parameters should be optimized to suppress the grating lobes.

1.2. Objective

The aim is to find a fixed element spacing and the resulting phase shift and amplitude distribution that allow for the suppression of grating lobes. This would result in a wider steering range and more efficient antenna. This must be achieved while keeping the half-power beamwidth (HPBW) as narrow as possible and maintaining a superior sidelobe level (SLL) and grating lobe level (GLL) compared to a uniformly spaced linear antenna array.

1.3. Scope

We will investigate what grating lobes are and how the parameters: element spacing, phase shift and amplitude distribution affect grating lobes. A Genetic Algorithm (GA) is implemented to optimise the parameters of the antenna array to suppress grating lobes. After a linear antenna array (LAA) spacing is selected the antennas will remain fixed and only the phase and amplitude of the signal entering the antenna can be altered to steer the beam. After different element spacing parameters were obtained each was tested for different steering angles to determine which performed best. The solution is then compared to the uniform antenna array of 0.5λ with the aim being to find parameters that perform better than the uniform linear antenna array (ULAA).

We did not investigate any antenna arrangements that are not linear such as planer antenna arrays. We will not investigate the feed system needed to implement the phase shift or amplitude distribution for a physical antenna array. Nor does the report describe how to build an antenna or antenna array. We will only investigate the theory to design an antenna array.

The report is set out as follows. Chapter 2 covers interesting related research that has been done as an attempt to solve the problem statement. Chapter 3 discusses antenna array fundamentals and how the Genetic Algorithm is designed to solve the problem. In chapter 4 the findings are documented and compared to current solutions of a uniformly spaced LAA spaced 0.5λ apart. The conclusion is drawn in the chapter 5. together with recommendations for future work.

Chapter 2

Literature Review

This chapter presents possible solutions to suppress grating lobes as given in the literature. It will focus on the influence non-uniform element spacing, phase shift and non-uniform amplitude distribution have on grating lobes. The focus will be on finding the best set of parameters for the array to increase steering direction to avoid or suppress grating lobes if they appear.

Cheng [2] showed that the configuration, element spacing, amplitude distribution and phase shift of the antenna elements affect the radiation pattern. This is also clear from inspection of the array factor (AF) equation (3.2)

In 1960 King et al. [6] presented a computational technique to design a non-uniformly spaced linear array. The designer can select an SLL and design the spacing accordingly. However, it was limited to only a certain number of Amplitude distributions. Unz. [7] was the first to explore the possibility of a non-uniform distribution of array elements when Unz presented a matrix relationship between the radiating pattern and array elements. The idea was however just stated in his short paper and never fully realised.

Harrington [8] explored the possibility of reducing SLL with non-uniform element spacing. He took inspiration from King et al. [6] who found certain non-uniform spacing that produced low SLL and outperform uniformly spaced elements. Harrington [8] managed to suppress the SLL without increasing the beamwidth of the main beam but stated: "It appears unlikely that a simple optimization procedure can be devised for nonuniform arrays because of the nonlinear character of the basic equations with respect to element position." - Harrington [8] But suggested that more work should be done on the topic.

Ridwan et al. [9] deployed a GA to design the optimal excitation amplitudes for a ULA and compared it to the Dolph-Chebyshev and Binomial amplitude distributions. [9]. Ridwan et al. produced amplitude tapering for a given set of antennas that outperformed a uniformly tapered LAA. Ridwan et al. did not explore the use of a GA for the spacing of the elements.

KhalilpourJafar et al. [10] in a recent publication publicized a novel algorithm to suppress grating lobes in a phased array when the separation between elements was around one wavelength apart. KhalilpourJafar et al. [10] showed significant SLL and GLL reduction for an eleven element phased array with up to 180-degree scan angle using randomly

spaced elements that creates nulls to cancel out grating lobes. It is important to note that randomly spaced array and non-symmetrically spaced arrays were found. This will be discussed later.

The shortcoming from the literature is that although GA is used together with Dolph-Chebyshev etc the beam is only steered straight forward or normal to the antenna array. Thus optimizing it for an antenna with no steering. The area of beam steering together with a non-uniformly spaced array leaves room for a lot of novel work to be done.

2.1. Array Geometry

From the study of other work done on the topic of the suppression of grating lobes it is clear that the spacing of the antenna elements is most important. We will consider the uniform array distribution, the non-uniform array distribution that is symmetrical around the mean and a random array distribution.

2.1.1. Uniformly Spaced Antenna Array

From the work done by Matsepe [11] it was shows that uniform arrays will have no grating lobes if the spacing between elements are uniform and less or equal to 0.5λ . However, beam steering is only possible to a certain angle that is known as the critical angle. Steering beyond this angle the side lobes levels become high enough to the point where they can be considered grating lobes.

The antenna array can be designed for a desired angle where the element spacing can be greater than 0.5λ with no grating lobes. When the distance decreases the constructive and deconstructive areas move further apart. Decreasing the distance between LAA elements result in suppressing grating lobes, for the distance between them move further apart, but increases HPBW. [11] Grating lobes are not suppressed using non-uniform Amplitudes such as binomial, Dolph-Chebyshev and Taylor. They suppress side lobes. Grating lobes bring side lobes. [11]

2.1.2. Symmetrical Non-Uniformly Spaced Antenna Array

The report will investigate the performance of an antenna array symmetrically spaced round the center point of the antenna arrays with spacing ranging from $0 - 2\lambda$. The spacing will be determined by the Genetic optimization algorithm. More about the GA in section 2.4.

2.2. Amplitude Distribution

Different amplitude distributions were recommended in literature where the binomial- and Dolph-Chebyshev distributions being the most common. It is mentioned by Matsepe [11] that changing the amplitude influences the SLL but has not effect on the grating lobes. The results in 4 supports this claim. However [10] showed that if you reduce the grating lobes to SLL you can suppress it further with amplitude tapering.

2.3. Grating Lobes

Grating lobes are unwanted lobes that appear when you steer the main beam beyond a critical angle that is determined by the array formation. These grating lobes are unwanted because they waste energy and cause interference from radiating stations at directions not in the intended direction. [12]

Grating lobes are as a result of the periodic nature of the AF as is clear from the Fourier Transform properties 3.1.

2.4. Genetic Algorithm

A Genetic Algorithm (GA) is an optimization algorithm that uses the concept of evolution and natural selection to solve complex problems. It is a very good approximation tool and does not always yield the best possible solution but can produce very good solutions relatively quickly. It uses crossover and mutation techniques to evolve a population of random solutions to produce the best approximation.

A population of size N of random solutions is created. Each solution is passed through a fitness or error function to determine how good the solution is. Once all N solutions have a fitness score a new population of size N is created and filled via a selection process that allows for cross over and mutation of the solution to occur.

The algorithm can be applied even if the fitness function is discontinuous, non-differentiable, stochastic or nonlinear.

2.5. Antenna Types

There are many different types of antennas Figure (2.1) for different applications. Omnidirectional antennas such as the monopole Figure (2.1a) and dipole Figure (2.1b) antennas can be made directional if placed in an array. There are directional antennas such as the Rhombic Figure (2.1d), Yagi Figure (2.1f) and Horn Figure (2.1e) antennas but with these comes the added complexity of antenna design and limitations during beam steering.

Since the most popular antenna used in antenna arrays is that of the half-wave dipole antenna Figure (2.1b) this project will use it during the calculations and simulation. This decision is based on the popularity of the half-wave dipole antenna, the simplicity of its design and keeping the focus on the spacing between the antenna and not on the type of antenna used.

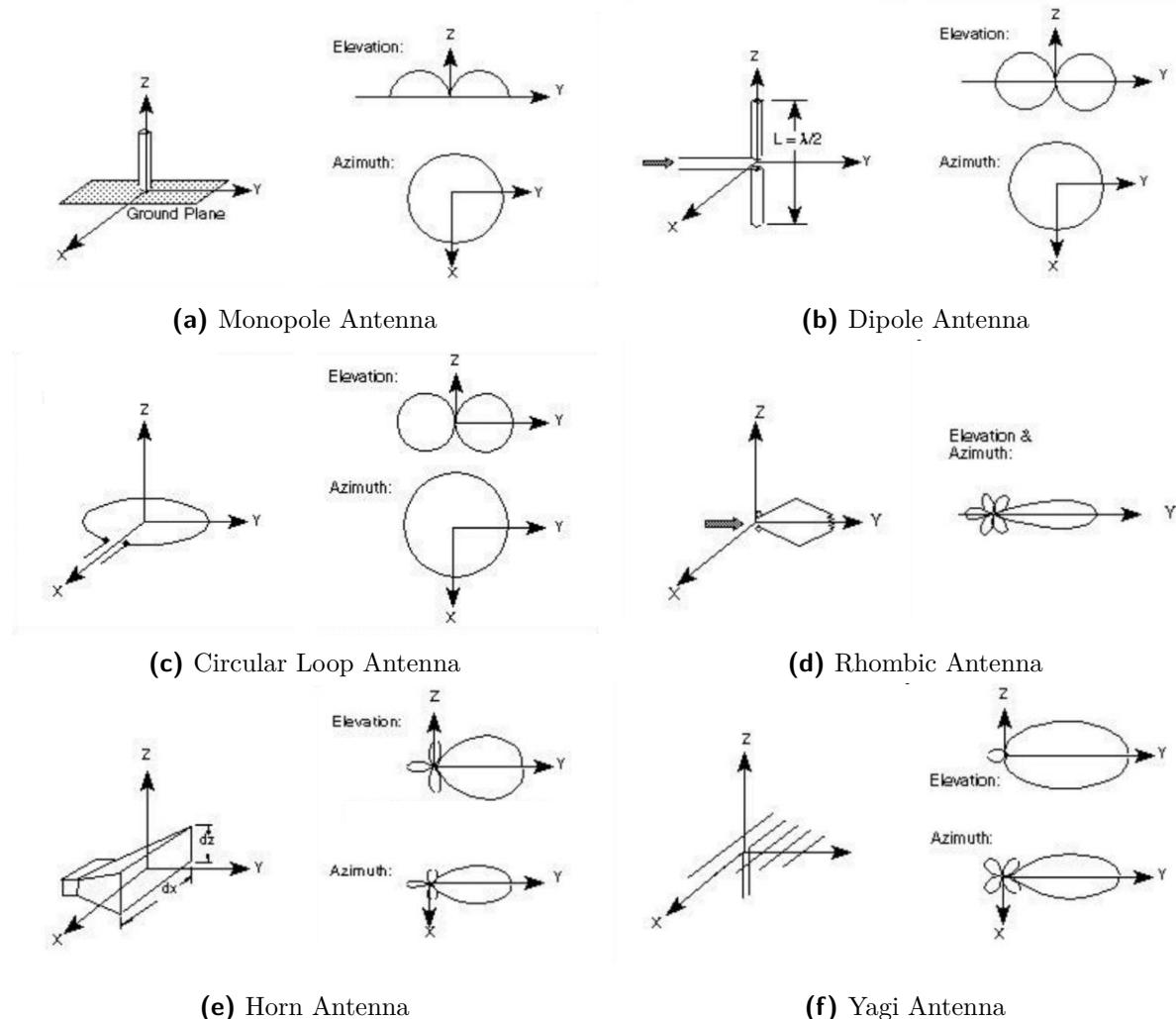


Figure 2.1: Antenna Types and Antenna Patterns [1]

Chapter 3

Design and Simulation

In this chapter, we will discuss the mathematical representation of the radiation pattern of an LAA also known as the array factor (AF) and how there is a relationship between the AF and the Fourier transform (FT). Once we understand the mathematics of the AF for an LAA we will design a GA to optimize the parameters: element spacing, phase shift and amplitude distribution to steer the beam off-boresight and while suppressing grating lobes.

3.1. Array Factor

The generalized AF is equation (3.1)

$$AF(\theta, \phi) = \sum_{m=0}^{N-1} I_m \exp(j[k\hat{r} \cdot \mathbf{r}'_m]) \quad (3.1)$$

Where $\hat{r} = \sin \theta \cos \phi \hat{\mathbf{x}} + \sin \theta \sin \phi \hat{\mathbf{y}} + \cos \theta \hat{\mathbf{z}}$. The LAA is spaced along the the z axis resulting in $\mathbf{r}'_m = md_m \hat{\mathbf{z}}$ yielding in $\hat{r} \cdot \mathbf{r}'_m = kd_m \cos \theta$

The Array Factor for a N element linear antenna-array far field radiation pattern is equation (3.2).

$$\sum_{m=0}^{N-1} I_m \exp(j\gamma_m) \quad (3.2)$$

$$\gamma_m = kd_m \cos \theta + \alpha_m \quad (3.3)$$

Where known as the wavenumber measures the number of wavelengths in a cycle. Each antenna can be modeled as a point source with amplitude I_m and phase shift α_m positioned a distance d_m away from the first antenna in the array with equation (3.4).

$$i_m = I_m \exp(jkd_m \cos \theta) \quad (3.4)$$

To achieve the maximum gain in the steering direction γ_m should equal to zero. This is achieved by equation (3.5)

$$\alpha_m = -kd_m \cos \theta_0 \quad (3.5)$$

This means that the maximum gain from the excitation of each antenna will be directed in a desired direction θ_0

All the antennas operate at the same operating frequency and to keep the calculations simple an operating frequency of 1 GHz was selected to match the speed of light at 300 m/s so that

$$k = \frac{2\pi f}{c} = \frac{2\pi}{\lambda} \quad (3.6)$$

The cosine function in equation (3.5) with amplitude equal to one will always be between the bands -1 and 1 as in equation (3.7).

$$|\cos \theta| \leq 1 \quad (3.7)$$

Solving the angles of equation (3.7) yields $[0 \leq \theta \leq \pi]$ or $[-\pi \leq \theta \leq 0]$. Only the region $[0 \leq \theta \leq \pi]$ will be considered since the two domains are symmetrical and this domain is known as the visible region.

Combining equation (3.2), (3.5) and (3.6) equation (3.8) can be derived.

$$\cos \theta - \cos \theta_0 = q \frac{\lambda}{d_{avg}} \quad (3.8)$$

This is an important equation because it shows that the AF equation (3.2) is periodic. There will be many main lobes but only $q = 1$ is considered the main lobe while every other q value that yield a real solution is considered a grating lobe. The equation can predict the location of the grating lobe by solving for θ for the desired steering direction θ_0 . It also shows that the only factor that influences how frequently the function repeats itself is determined by the average spacing distance d_{avg} . In other words we can see from equation (3.8) that if we make the spacing d_{avg} smaller the distance between the grating lobes increases and if the spacing d_{avg} is increased the distance between the grating lobes decrease.

3.2. Fourier Transform and the AF

The AF equation (3.2) is the sum of N antenna elements shifted by some distance d_m from the first antenna element. This is also the equation of the Discrete Fourier Transform (DFT). This means the AF is equal to the Fourier transform of the antenna elements. As shown in equation (3.4) each antenna element can be modelled as a point source with a frequency shift. Let us assume that each element is uniformly excited and that the spacing between the antennas are uniform. Then the antenna-array can then be seen as a sampled rectangular window function in Figure (3.1c). From our Fourier theory, we know that a rectangular window function transforms into a Sinc

function (Figure (3.1d)) when the Fourier transform is applied [13]. The main beam of the sinc function is the main beam of the AF and the ripples of the Sinc function are the side lobes that come with the main beam. However, when we sample the rectangular window function we get an impulse train equalling that of the N antennas in Figure (3.1c). The Fourier transform of a sampled rectangular window function is a sinc function convolved with a Dirac comb function in Figure (3.1f). These extra main beams are the GL we want to suppress. See the graphical representation Figure 3.1

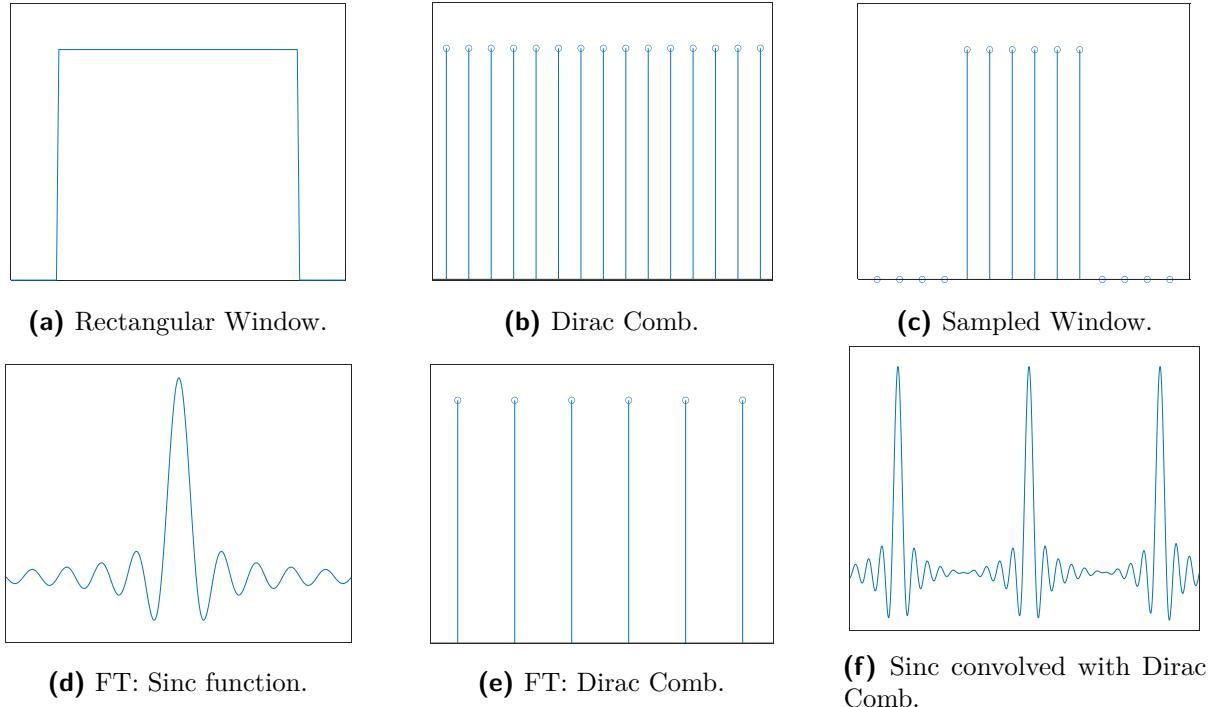


Figure 3.1: Visual representation of the Fourier transform of a LAA.

Figure (3.1a) shows the rectangular window with a width equal to N. The sample function can be modelled as a Dirac comb function with the period $T = \frac{\lambda}{d}$ multiplied by a rectangular window. From our FT understanding, the AF will be a sinc function convolved with a Dirac comb function equation (3.9).

$$AF = rect(N) * \sum_{-\infty}^{\infty} \delta(x - md) \quad (3.9)$$

In our case it will be in the visible domain which is between 0° to 180° .

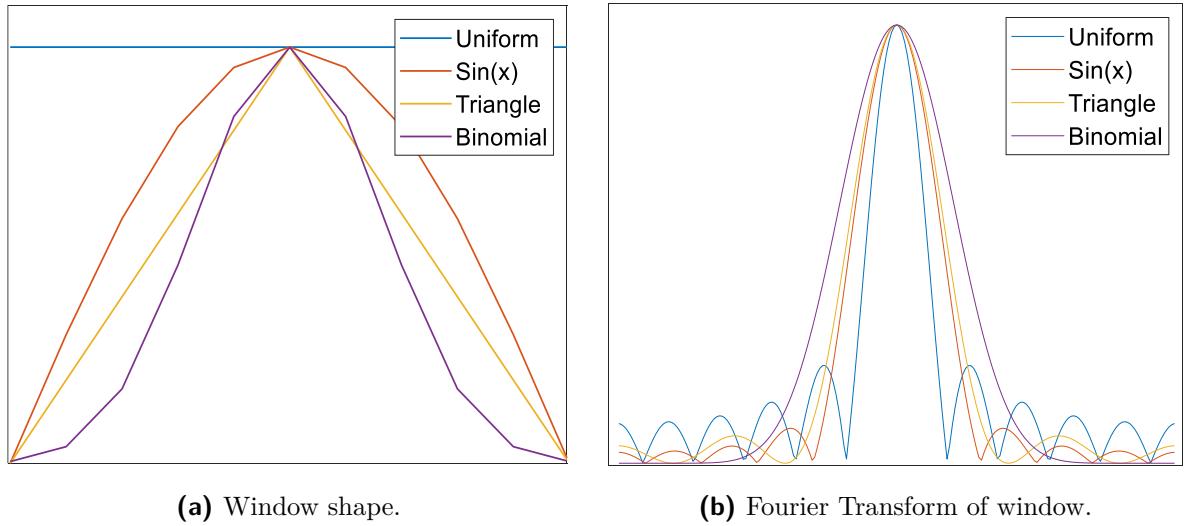
$$AF = \frac{\sin \frac{\pi N}{\lambda}}{\frac{\pi N}{\lambda}} * \sum_{0^\circ}^{180^\circ} \delta(u - \frac{m\lambda}{d}) \quad (3.10)$$

This periodicity of impulses is the cause of grating lobes. The first sinc beam being the main beam and every consecutive sinc beam being a grating lobe. The position of the m_{th} beam or grating lobes is represented by $m\frac{\lambda}{d}$. This again shows that decreasing the d

will increase the distance between the grating lobes.

The Nyquist criteria states that the sampling frequency must be at least twice as large as the highest frequency you want to sample to be able to fully reconstruct the signal without aliasing [13]. Since the antenna array can be seen as a sampled rectangular window with different excitation levels the sampling frequency must be $d \leq 0.5\lambda$ to avoid aliasing. From digital signal sampling, we know that this holds true even if we sample at a random interval i.e. the $f_{avg} \geq 2B$ to avoid aliasing. This indicates that a randomly spaced array would need to have an average spacing of 0.5λ if grating lobes are to be suppressed.

From our FT theory understanding, we know that changing the window shape (Figure (3.2a)) has an effect on the width and ripples of the Sinc function (Figure (3.2b)) or the side lobes [12] [13].

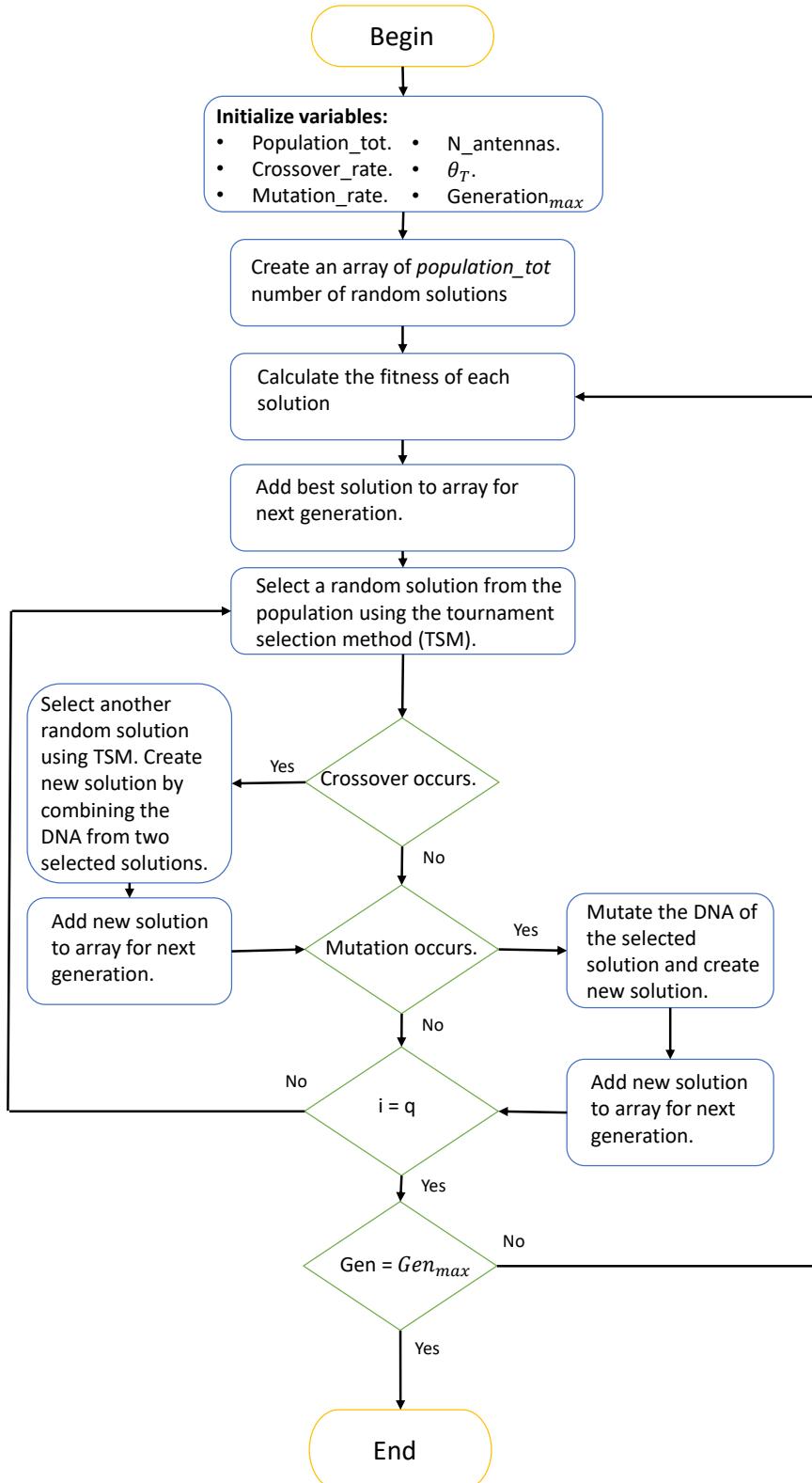


(a) Window shape.

(b) Fourier Transform of window.

3.3. Genetic Algorithm

The GA will be structured as follows.



(a) GA flow chart

The first step of the GA is to initialize the different parameters for the number of antennas, N . The number of random solutions the algorithm will work with is known as the population or *population_tot*. Each of the solutions in the population will be given a fitness score. This score is the error of the solution which means that a lower score indicates that the solution is better relative to the other solutions in the population. The GA will aim to minimize the fitness score until it reaches an absolute minimum or equilibrium. The best solution of the current generation is automatically added to the array for the next generation. During each generation the solutions will have a probability to undergo crossover (*crossover_rate*) and mutation (*mutation_rate*) before getting added to the next generation. Crossover allows the best solutions to share DNA (parts of their solution) to produce a potentially better solution for the next generation. Mutation is responsible to add some randomness to the solutions to help the algorithm get out of local minimums and reach the global minimum. This process is repeated until the current generation reaches the max number of generations decided by the user.

3.3.1. Fitness score

The most important part of the GA is the fitness function. It describes how good a solution is. In the case of the antenna array it takes the element spacing, phase shift and amplitude distribution of each antenna as input parameters to calculate the AF. It then uses the AF to calculate a fitness score for the LAA.

Equation (3.11) calculates the ratio between the maximum gain at a desired angle θ_T to the average gain over the entire visible space. This is also known as the directivity equation. It is used to give an indication how well an antenna is performing.

$$D_o = \frac{\pi |AF_{max}|^2}{\int_{\pi}^0 AF} \quad (3.11)$$

Removing the constant π and taking the inverse of the directivity over the visible space the following fitness function is derived equation (3.12).

$$fitness = \frac{\sum_0^{\pi} AF(\theta)}{AF(\theta_T)^2} \quad (3.12)$$

The GA tries to minimize the fitness function as far as possible. The numerator in the fitness function is wasted energy and should ideally be zero. The denominator is the gain we want to maximize. Maximizing the denominator or minimizing the error/numerical would minimize the fitness score. It is unlikely that the GA will find the absolute minimum value but it will obtain a really good one. These fitness scores are then used during the selection process.

3.3.2. Selection

Two popular selection techniques were considered namely tournament selection and roulette selection. Roulette selection scales the probability of a solution being selected at random by its fitness score relative to all the other solutions fitness scores. This means the better the solution is relative to the population the more likely it is that it will be selected from the population for crossover and/or mutation. This creates a bias towards the better solutions and limits the diversity in the overall population increasing the risk of getting stuck in a local minimum. Tournament selection selects four random solution from the population with equal probability and then selects the two solutions with the best fitness score from the four for crossover and/or mutation. Tournament selection is therefore implemented due to it being less biased towards the best solution.

3.3.3. Genome

Each member in the population is known as a Genome and describes a potential solution for the main beam a LAA of size $N = 6$ steered to a desired direction. The solution of the genome is known as its DNA. Each genome has a fitness score, an array of size $N - 1$ describing the spacing between the antenna elements, an array of size N describing the phase shift of each antenna element and an array of size N describing the amplitude distribution of each antenna element.

3.3.4. Crossover

Crossover is a method that uses known solutions, (two parent solutions) to produce a new solution (new genome) by combining the DNA of the parent solutions. During every selection process, there is a probability that crossover can occur or a crossover rate. When no crossover occurs the selected solution from the population is immediately copied over to the next generation.

When combining the DNA of the two parent arrays as illustrated in Figure (3.4) one must introduce a weighting factor. The weights for α and β can be selected freely but $\alpha + \beta = 1$. This ensures that the new solution that is generated is a valid solution that falls within the expected range. For the inter element spacing the value must fall between 0.1λ and 1λ . The phase shift must fall between 0° and 360° . The amplitude distribution must fall between 0 and 1. If an α and β value of 0.5 is chosen the resultant solution is simply the midpoint of each array index of the two parents.

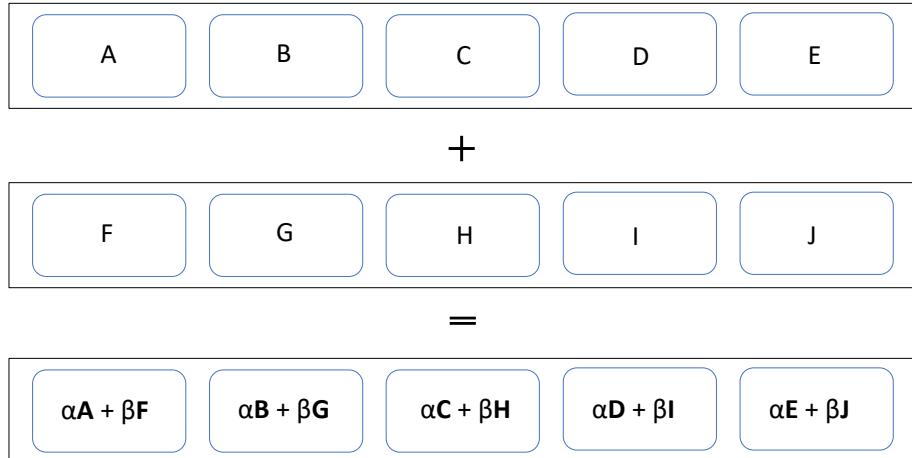


Figure 3.4: Crossover process

3.3.5. Mutation

During every selection process, there is a probability that a mutation can occur or a mutation rate. During this process, the selected solution will undergo a random change to its DNA. Mutation is illustrated in Figure (3.5) using a binary array of length $N = 5$. The fourth index is randomly selected and the value is changed from a 1 to a 0. The reason for mutation is to help the algorithm not get stuck on a local minimum/maximum solution. This is achieved by introducing some randomness to the solution. The new solution is then added to the next generation.

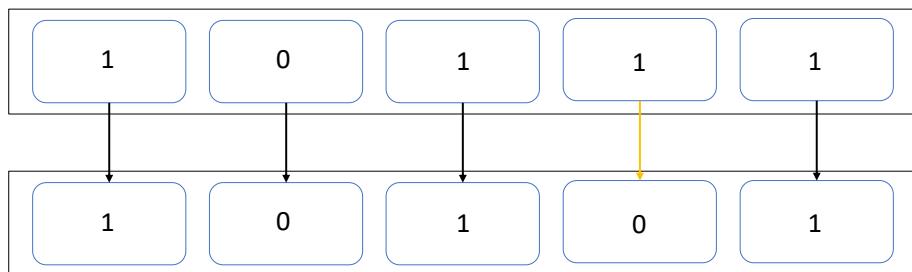


Figure 3.5: Mutation process

3.4. Chapter Summary

In this chapter, we investigated the radiation pattern of an LAA and expressed it mathematically with the AF. Using the array factor and our knowledge of the FT we could prove that it's periodic and that this fact gives rise to GL's. Using the AF and the directivity function of antennas we could design a GA to optimize the spacing of the antennas and digital parameters to steer and suppress grating lobes.

Chapter 4

Results

It was decided in an earlier section to use a half-wave dipole antenna. Table (4.1) shows the parameters used to model the dipole antenna in a MATLAB package called Antenna Toolkit.

Variable	Value
frequency	3GHz
c_0	$3 \times 10^8 m/s$
dipole radius	0.001m
λ	1m

Table 4.1: Antenna Setup

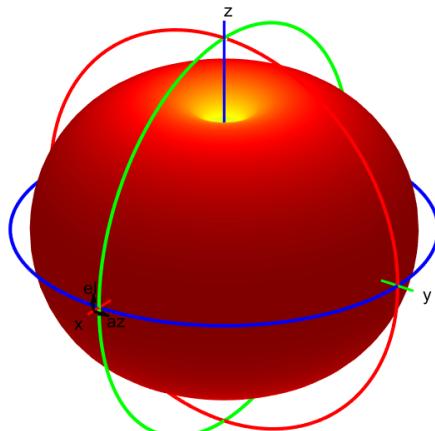


Figure 4.1: Half-wave dipole antenna radiation pattern.

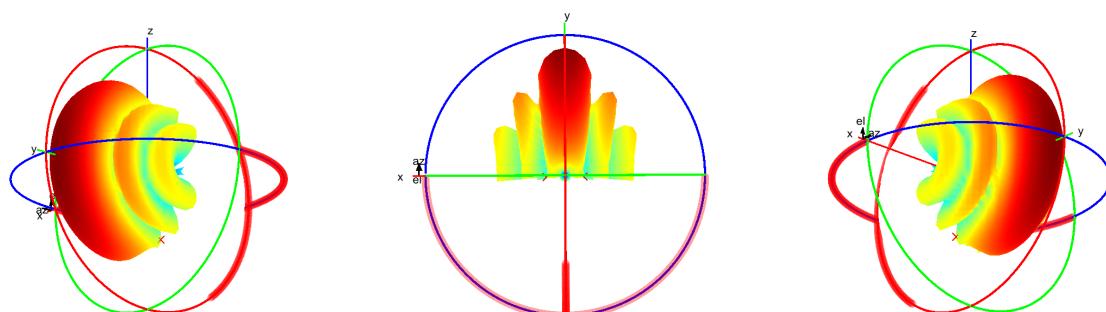


Figure 4.2: 3D radiation pattern for 6 element LAA.

4.1. Uniform linear Antenna Array Results.

To design a non-uniform LAA we first want to understand how well the uniform LAA performs. We are going to use the uniform LAA as a baseline to measure how well our GA performs so that we can then use the GA to find the best non-uniformly spaced LAA. The GA is used to optimize the element spacing, amplitude distribution and phase shift for the radiation pattern steered boresight and to 70° from boresight. The best results are document in this section.

4.1.1. Amplitude Distribution

From our research in the literature review Chapter 2 it was stated that the amplitude distribution only affect the SLL and not the grating lobes. When we look at the Fourier transform of different window functions in Figure (3.2b) it is clear that the SLL is reduces while the HPBW is increases. Using the GA to optimize the amplitude distribution for a LAA spaced 1λ apart it is shown that the SLL were reduced by 21.21 dB compared to the uniform distribution with only a slight increase in the HPBW.

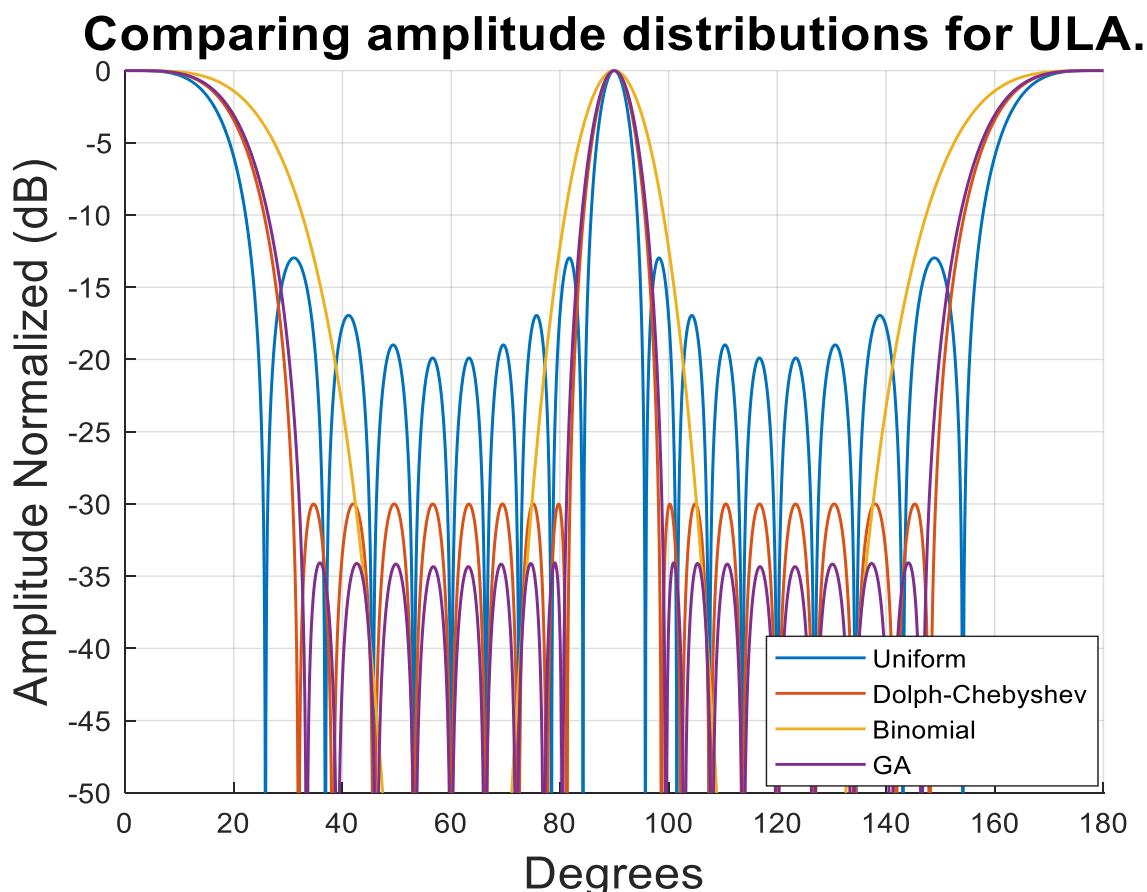


Figure 4.3: Uniform vs Dolph-Chebyshev vs Binomial Amplitude vs GA amplitude distribution.

4.1.2. Uniform linear Antenna Array Beam Steering

Equation (4.1) is used to steer the radiation pattern to a target direction θ_T .

$$\Delta\delta = \frac{360^\circ d \sin(\theta_T)}{\lambda} \quad (4.1)$$

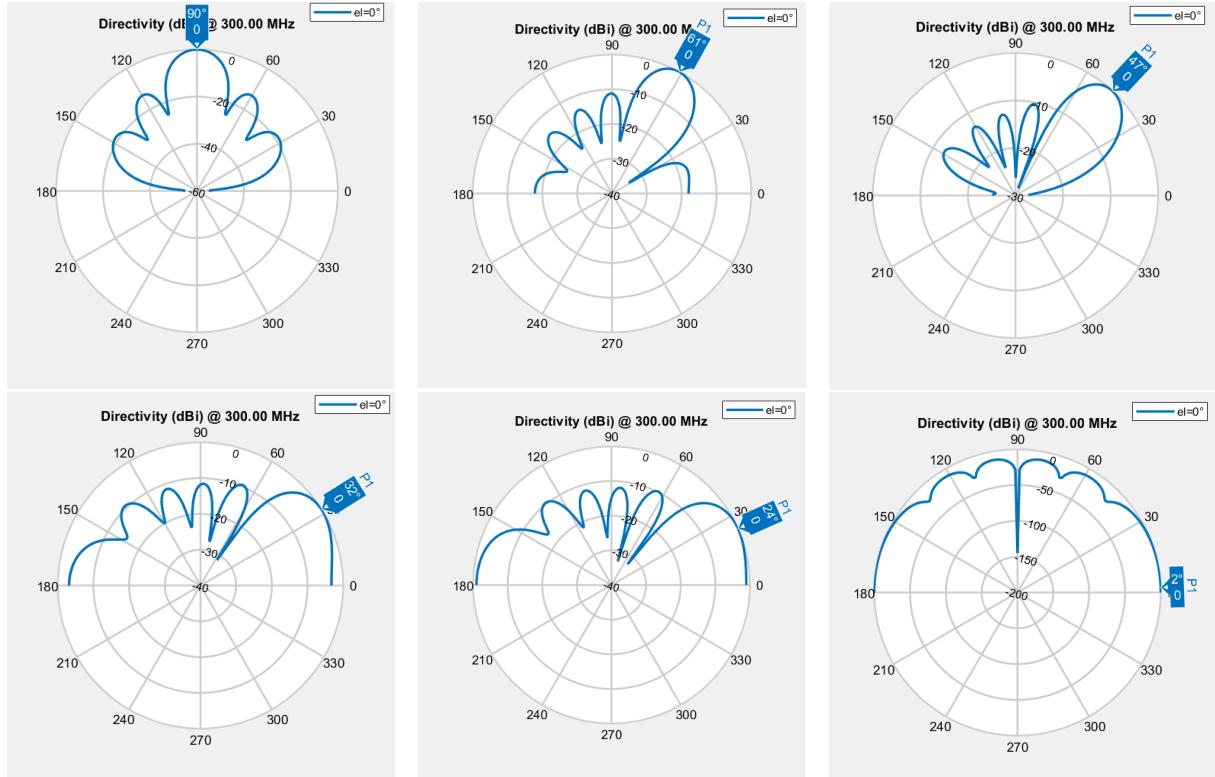


Figure 4.4: Beam steering (with equation (4.1)) with a uniform LAA of 6 antennas elements spaced 0.5λ apart.

θ_T	HPBW	SLL dB	$\Delta_{Directivity}$	Measured	θ_{err}	$\Delta\delta$
90°	18°	-13.29	4.935	90°	0°	0°
60°	20°	-11.21	4.064	61°	1°	90°
45°	23°	-10.35	3.52	47°	2°	127.28°
40°	25°	-9.913	3.09	43°	3°	115.7°
30°	32°	-4.714	2.412	34°	4°	169.14°
27°	36°	-3.191	2.293	33°	6°	95.38°
20°	+40°	-1	2.144	24°	4°	155.88°
10°	+o	0	2.094	0°	-	177.27°

Table 4.2: The performance of a Uniform LAA spaced 0.5λ apart steered using equation (4.1).

The results obtained for steering a 6 antenna LAA with uniform amplitude distribution

spaced uniformly 0.5λ apart to different target directions using eq. (4.1) is shown in figures 4.4 and table 4.2.

As the beam is steered from boresight to 63° from boresight the HPBW increases from 18° to 36° while the SLL shifts from -13.29 dB to -3.191 dB . When the beam is steered beyond 36° the side lobe increases beyond -3.191 dB and turns into a grating lobe. This means the LAA has a steering range $\pm 63^\circ$ from boresight.

When we compare the theoretical steering target direction to the measured direction achieved during simulation a small steering error occurs. This means when we use eq. (4.1) to steer the beam to 27° the beam is measured to be at 33° . This θ_{err} becomes larger (0° to 6°) the further away from boresight the beam is steered. This means the actual steering range achieved by the LAA is $90^\circ - 33^\circ = 57^\circ$ symmetrically around boresight.

These results will be the baseline for the rest of the simulations to either match or improve upon.

4.1.3. GA Optimizes Phase Shift for ULA.

The GA is used to steer the beam of a uniformly spaced LAA with uniform amplitude distribution to different target directions and is then compared to the steering formula eq. (4.1) to measure its performance.

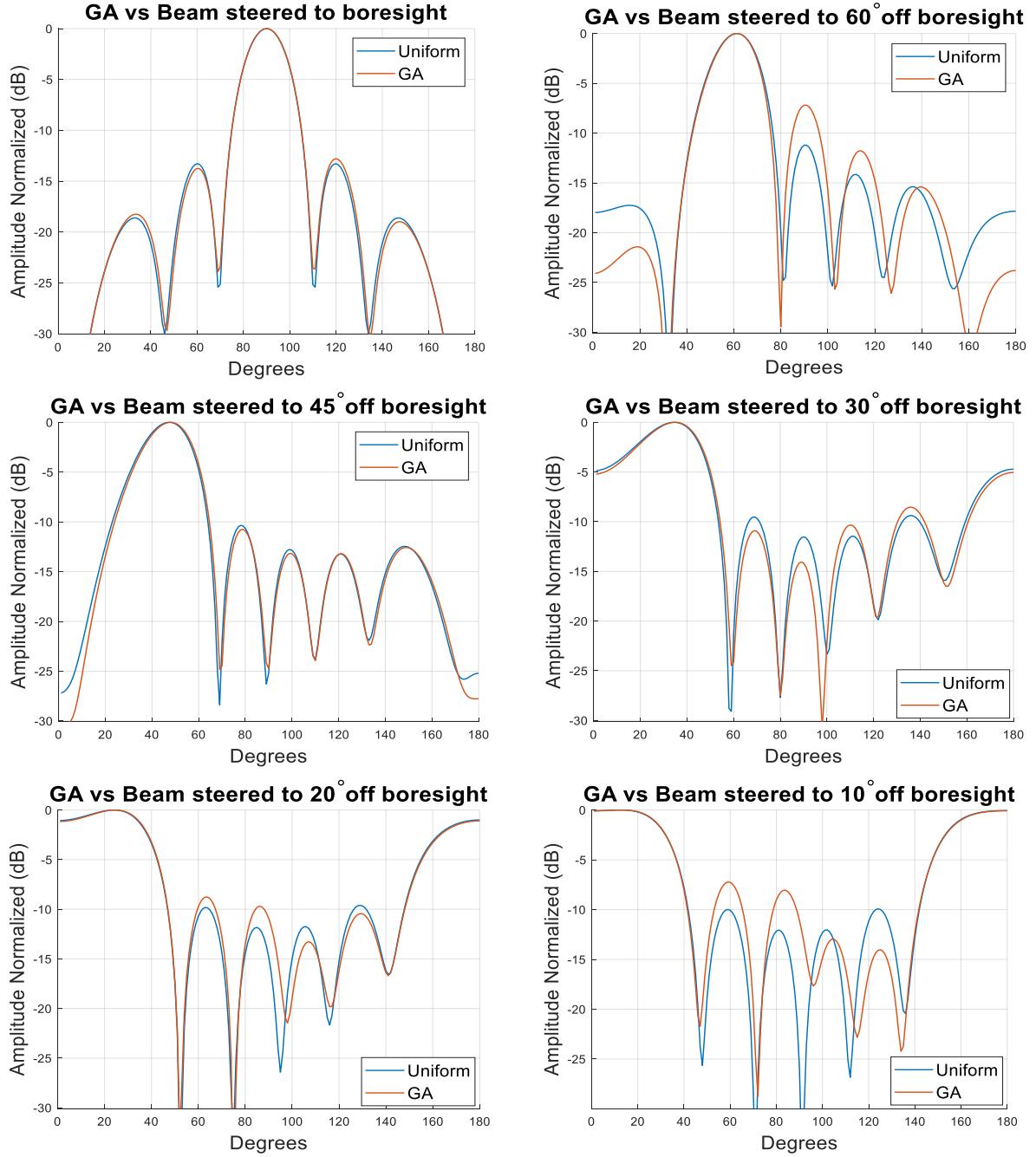


Figure 4.5: The AF for Uniform LAA spaced 0.5λ apart steered using the GA.

θ_T	HPBW	SLL	fit	GA fit	Δ_{fit}	Measured	Calculated	θ_{err}
90°	18°	-12.79	6.406	6.36	-0.046	90°	90°	0°
60°	20°	-7.195	7.931	7.36	-0.571	61°	60°	+1°
45°	23°	-10.76	8.525	8.404	-0.121	48°	45°	+2°
30°	32°	-4.714	12.374	12.33	-0.044	34°	30°	+4°
20°	+39.5°	-1.097	13.566	13.574	0.008	24°	20°	+4°
10°	+0°	0	13.866	13.997	0.131	13°	10°	+3°

Table 4.3: The performance of a uniform LAA spaced 0.5λ apart steered using the GA.

When steering the radiation pattern of the LAA using the GA to different target directions between boresight and 80° from boresight the GA matches the steering algorithm eq. (4.1) almost exactly, see Figure (4.5).

The GA aim is to minimize equation (3.12) which reduces the error of a particular solution at a desired target direction. Applying the fitness function to both the solution determined by the steering equation and the solution determined by the GA yield fitness scores as seen in table 4.3 in the *fit* and *GA fit* column. The biggest difference in fitness score is $\frac{0.571}{7.931} = 7.2\%$ which is also an outlier (more about this result in the next paragraph) with the second biggest being $\frac{0.121}{8.525} = 1.4\%$. The GA algorithm managed to obtain a phase shift for each antenna in the LAA that produced a fitter solution than that of the steering algorithm, meaning the fitness score is lower for the GA solution than that of the steering equation using equation (3.12).

Something interesting happens at 60° when the GA produces a solution that is 7.2% fitter than that of the uniform LAA. The SLL is clearly higher for the GA solution than for the uniform LAA however, the fitness score is 0.571 points better for the solution of the GA. This means the fitness function converges to a lower value than that of the solution of the steering equation for a uniform LAA even though the SLL are higher. This is as a result of the fitness function and its limitations rather than the performance of the GA. It indicates that the fitness function needs to be adjusted by adding more constraints to prioritise suppressing the highest SLL first before suppressing the average SLL. For our application the fitness function performs well enough within a reasonable error margin of well below 10% and will be used to steer the beam for a non-uniform LAA.

4.2. Uniformly spaced array

The only parameter that has an affect on the grating lobes is the spacing between the antenna elements of the LAA. To understand how the element spacing influences the grating lobes the radiation pattern is simulated for different uniform spacing ranging from 0.5λ to 5λ .

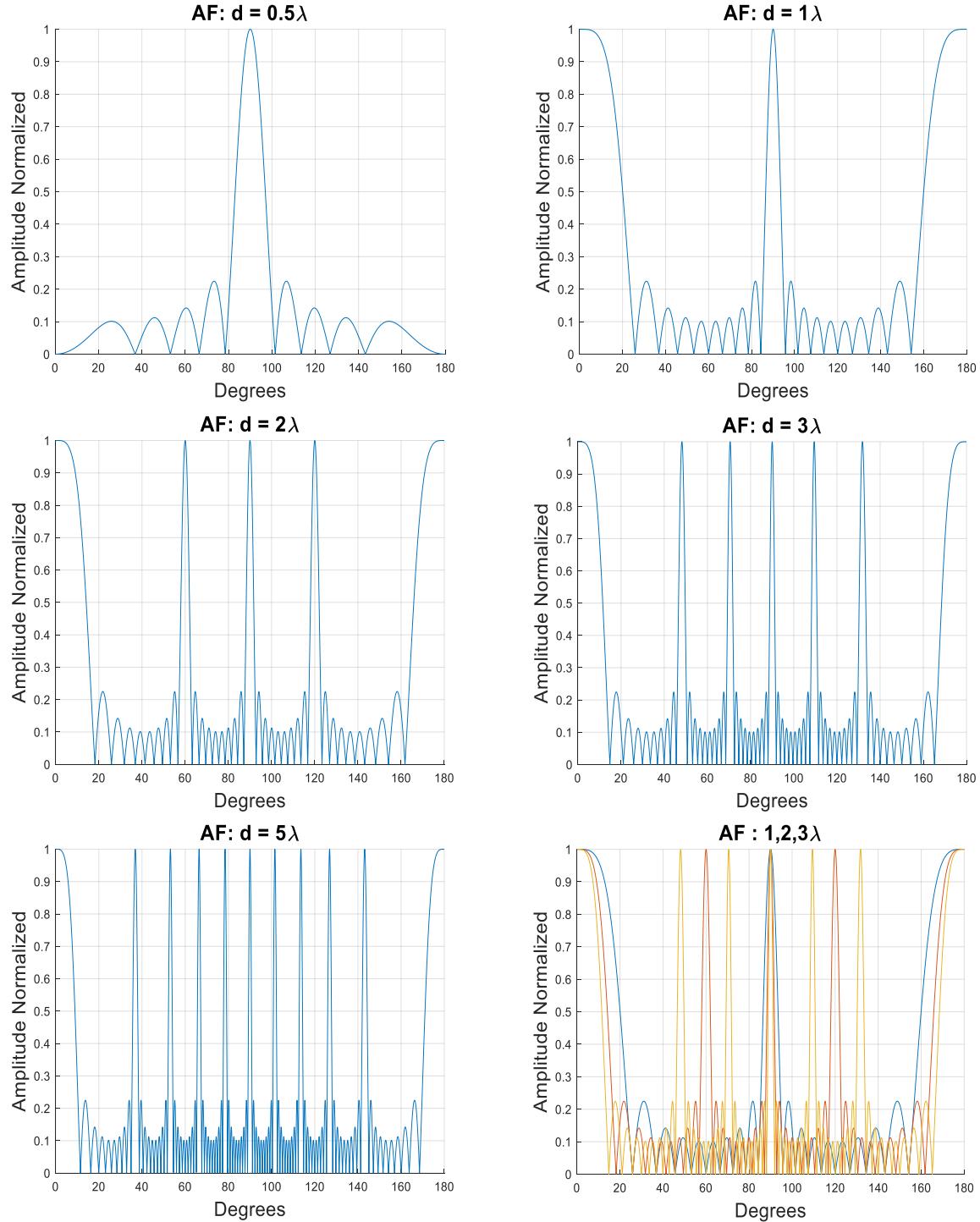


Figure 4.6: The AF for a uniform LAA spaced from 0.5λ to 5λ apart.

Uniformly Spaced	GL $k = 1$	Predicted	GL $k = 2$	Predicted	GL $k = 3$	Predicted
0.5λ	NONE	-	NONE	-	NONE	-
1λ	0°	0°	NONE	-	NONE	-
2λ	60°	60°	0°	0°	NONE	-
3λ	70.54°	70.528°	48.18°	48.189°	0°	0°
5λ	70.46°	78.463°	66.42°	66.421°	53.13°	53.13°

Table 4.4: Mathematical predictions vs AF calculation for the location of grating lobes.

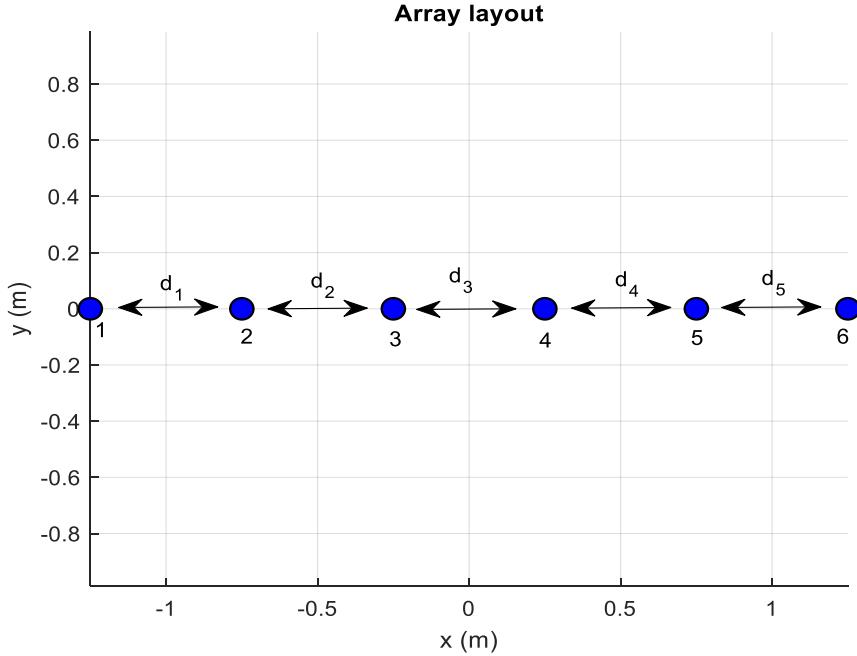


Figure 4.7: Six antennas separated by a constant value $d = x\lambda$

The position of the grating lobes are predicted by the equation $-1 \leq \arccos(k_d^\lambda) \leq 1$ for all the value of k resulting in a real solution. The solution must be real for only real solution will appear in the visible space $0^\circ \leq \theta \leq 180^\circ$.

The results for the predicted values compared to the simulated values for different uniform LAA's are in table (4.4) and is accurate to within 1%. This means we have shown that we can mathematically predict where the grating lobes will appear for a given uniform spaced LAA and the affect the spacing has on the grating lobes.

A key takeaway from Figure (4.6) is that as the spacing between a uniform LAA increases the spacing between the grating lobes becomes smaller and visa versa. Another observation that can be made from Figure (4.6) is that as the spacing between the antenna elements increase the HPBW becomes smaller. This means there is a trade off between the HPBW and the spacing between grating lobes.

The following Figures (4.8), (4.10) and (4.12) show the 3-dimensional radiation pattern of the LAA with different uniform spacings.

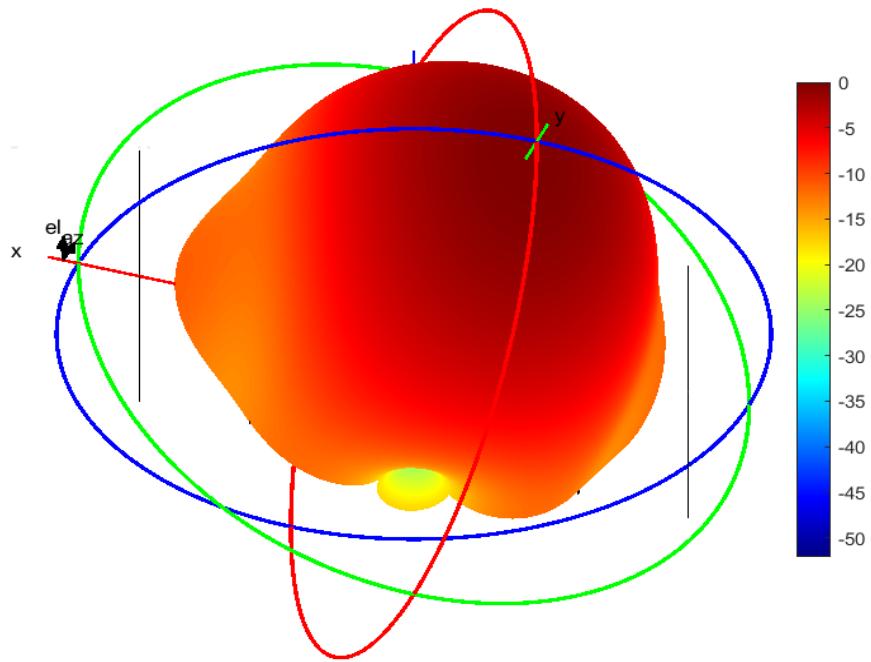


Figure 4.8: Uniform LAA with 6 elements spaced 0.17λ apart.

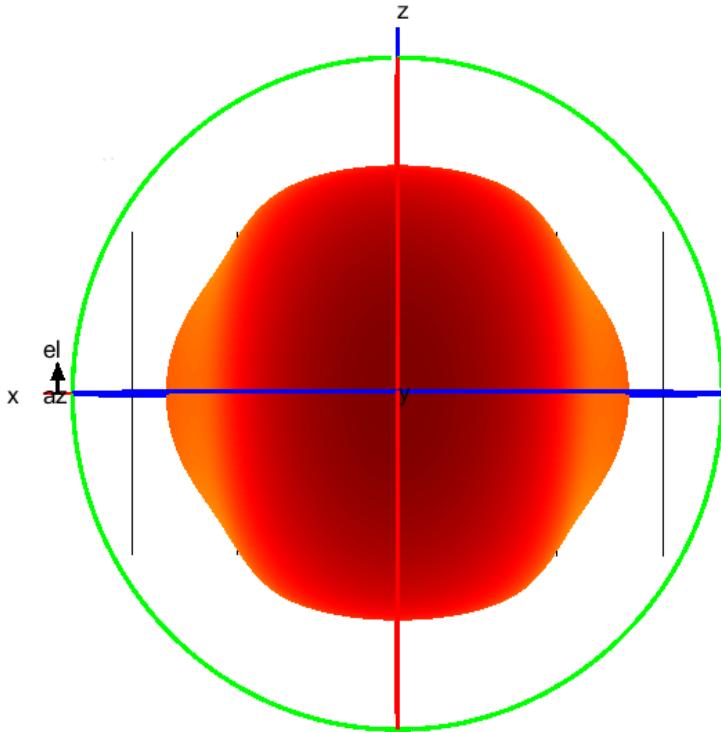


Figure 4.9: Uniform LAA with 6 elements spaced 0.17λ apart top view.

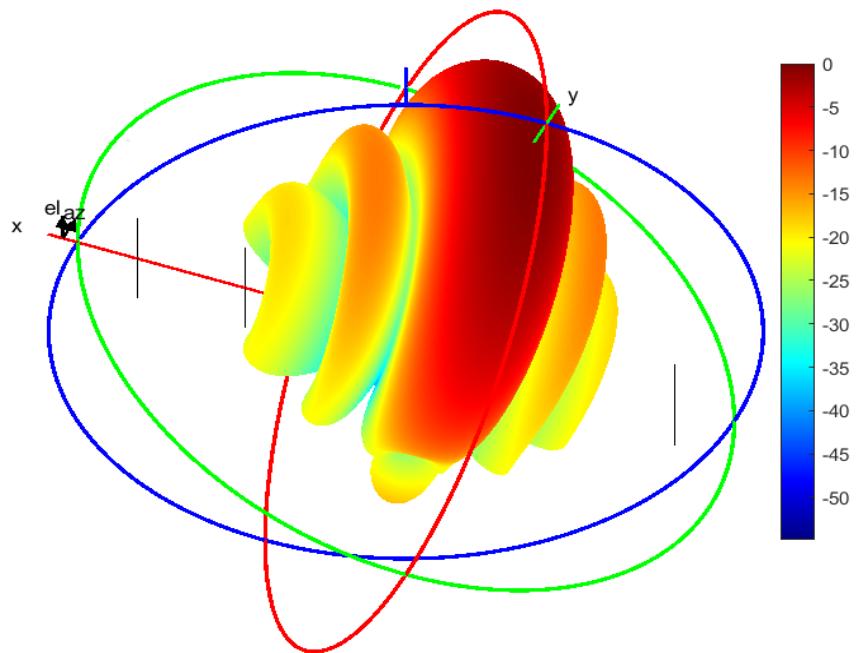


Figure 4.10: Uniform LAA with 6 elements spaced 0.5λ apart with no grating lobes.

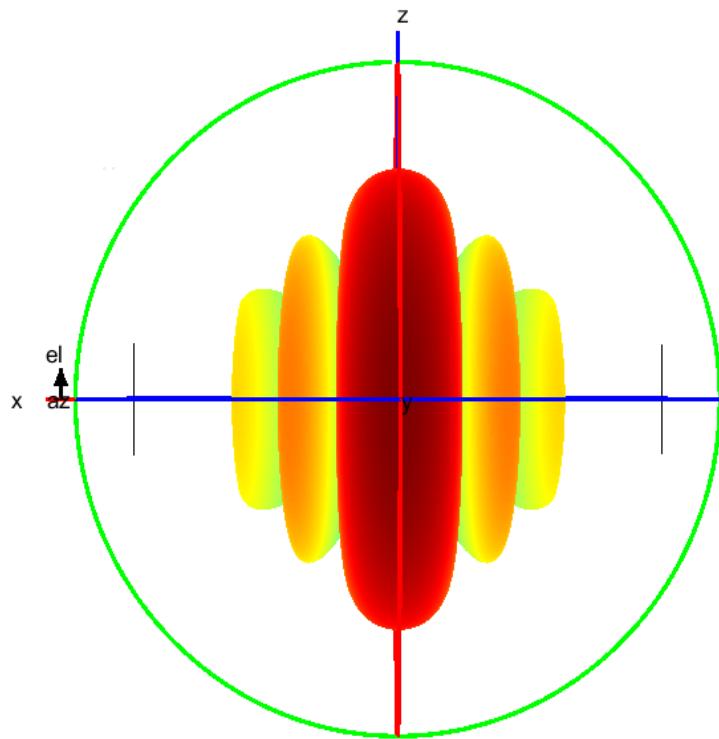


Figure 4.11: Uniform LAA with 6 elements spaced 0.5λ apart top view.

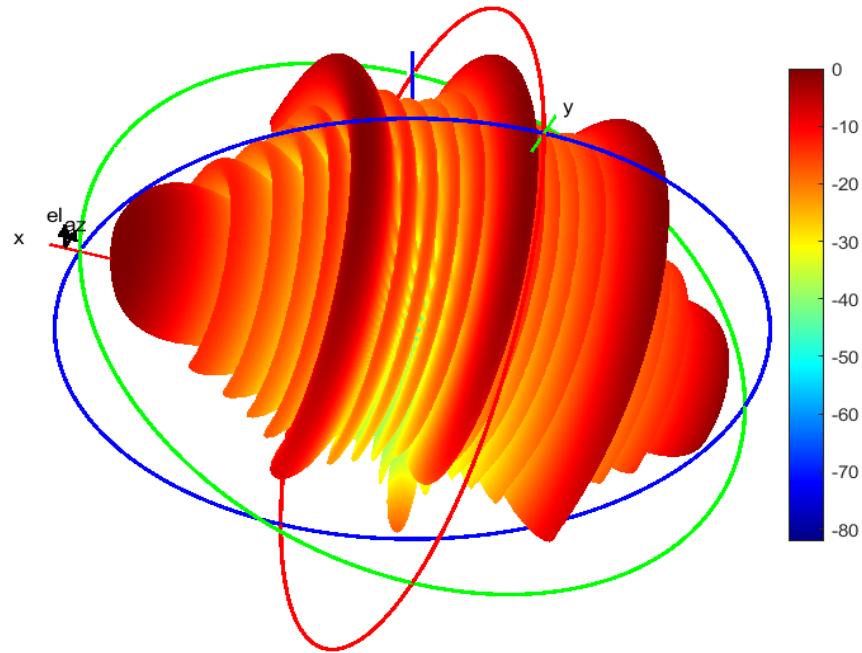


Figure 4.12: Uniform LAA with 6 elements spaced 2λ apart.

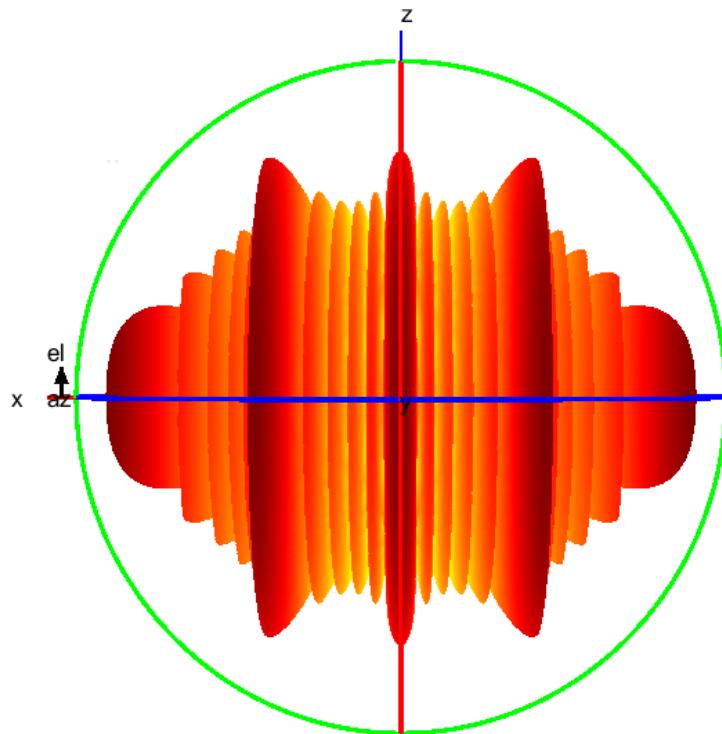


Figure 4.13: Uniform LAA with 6 elements spaced 2λ apart top view.

In Figure (4.8) the LAA is spaced 0.17λ apart and has no grating lobes and no side lobes but has a very large HPBW as predicted. When the LAA is spaced 0.5λ apart as in Figure (4.10) the uniform LAA (the current best solution) has some side lobes and no grating lobes. Only when the LAA is spaced is spaced 2λ apart as in Figure (4.12) we see

two grating lobes appear that are positioned symmetrical around the main beam.

The main beam, side lobes and grating lobes appear as rings around the x axis. This is due to the fact that the antenna array is 1-dimensional and that the dipole antennas radiate in all directions equally. Resulting in constructive wave in the shape of a ring around the x axis.

By taking a slice of the x, y plane of the 3D plots we can plot the graphs in Figure (4.6).

4.3. Symmetrical non-uniformly spaced array

Using the GA three solutions (see table 4.5) were obtained that suppress the grating lobe that appears when a uniform LAA spaced 0.5λ apart is steered beyond the critical angle (CA). The performance of the three solutions (S_1 = solution 1, S_2 = solution 2, S_3 = solution 3) is compared to the performance of the uniform LAA spaced 0.5λ in Figure (4.14) and documented in Table (4.6).

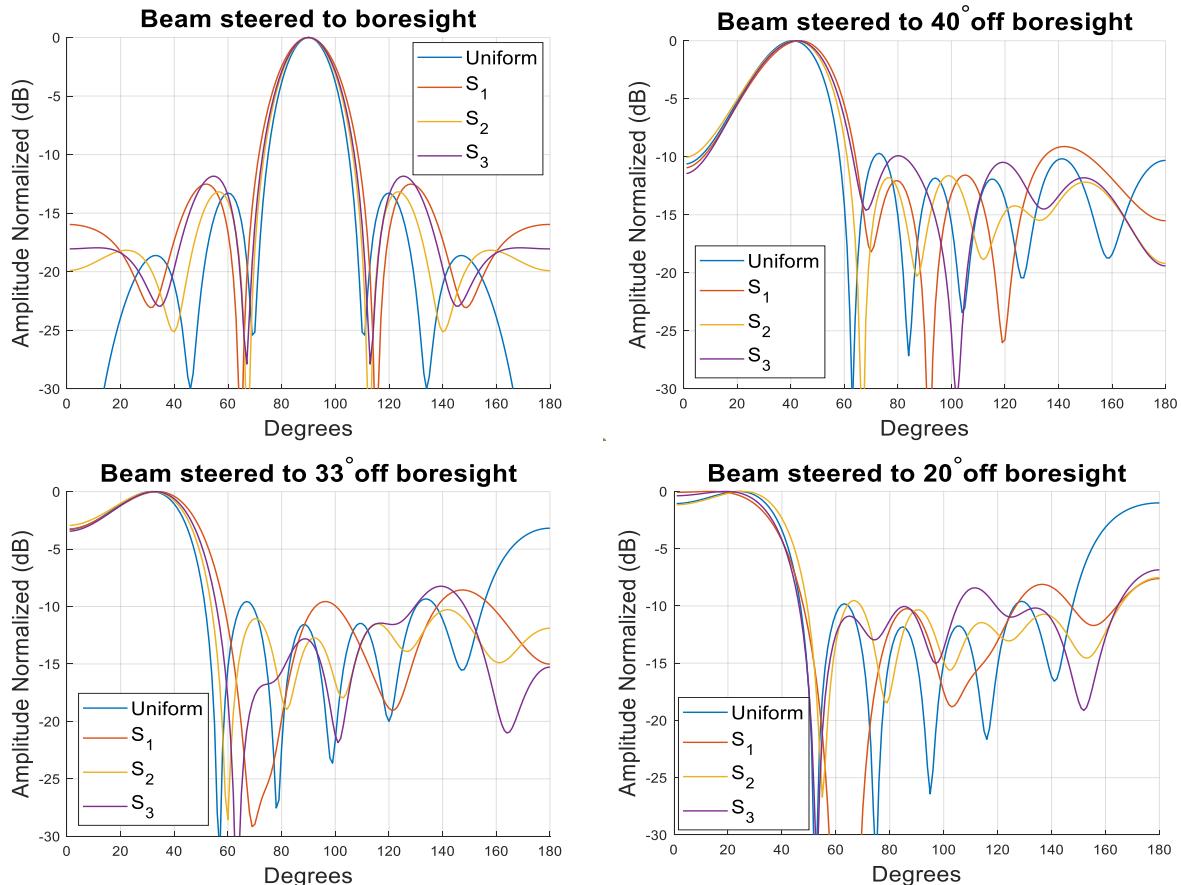


Figure 4.14: Steering beam from boresight to different target directions.

Solution	Inter Element Spacing λ
ULA	[0.5 0.5 0.5 0.5 0.5]
S_1	[0.3 0.475 0.52 0.475 0.3]
S_2	[0.425 0.475 0.5 0.475 0.425]
S_3	[0.35 0.48 0.56 0.48 0.35]

Table 4.5: Element Spacing that yield best grating lobe suppression.

θ_T	Solution	HPBW	SLL	Δ_{HPBW}	$\Delta_{SLL} (dB)$	$\Delta_{Directivity}$	$\Delta_{\theta_{err}}$
90°	S_1	22°	-12.52	+4°	+0.77	-1.187	0°
90°	S_2	20°	-13.19	+2°	+0.1	-0.719	0°
90°	S_3	20.5°	-11.84	+2.5°	+1.45	-1.024	0°
40°	S_1	29.5°	-9.134	+4°	+0.568	-0.2527	1.5°
40°	S_2	28°	-11.62	+2.5°	-1.918	-0.0075	1°
40°	S_3	28.5°	-9.907	+3°	-0.205	-0.1675	1°
33°	S_1	43°	-8.563	+7°	-5.372	+0.0749	1°
33°	S_2	46°	-10.31	+10°	-7.119	+0.2146	1.2°
33°	S_3	39.5°	-8.236	+3.5°	-5.045	+0.1326	1.1°
70°	S_1	37°	-7.613	-2.5°	-6.613	+0.3198	0°
70°	S_2	41°	-7.523	+1.5°	-6.523	+0.2284	0°
70°	S_3	37.5°	-6.846	-2°	-5.846	+0.2307	0°

Table 4.6: Performance of non-uniformly spaced solutions.

When comparing the solutions in Figure (4.14) the first observation we make is that the HPBW is slightly wider for the three solutions obtained by the GA compared to the uniform LAA at all target directions. When the main beam is directed boresight (90°) the solutions obtained have a HPBW 2° to 4° (or 10% to 20%) wider with SLL 0.1 dB to 1.45 dB (or 5.7% to 10.9%) higher than the uniform LAA. With the SLL being higher and the HPBW being wider the directivity of the solutions are $S_1 = -1.187$, $S_2 = -0.719$ and $S_3 = -1.024$ less directive than the uniform LAA.

When we steer the beam to the CA of 33° or 57° from boresight the solutions obtained by the GA are superior to that of the uniform LAA. At the CA the HPBW ranges from 39.5° to 46° which is 3.5° to 10° (9% to 27%) wider than the uniform LAA. The SLL of the uniform LAA is -3.191 dB with the solution with the worst SLL being at -8.236 dB resulting in a 5.045 dB, (61.2%) reduction of the growing grating lobe. Though the HPBW is still wider than that of the uniform LAA the superior SLL allows the directivity of all the solutions to be better than that of the uniform LAA.

When the beam is steered beyond the CA the uniform LAA officially has a grating

lobe however, the solutions obtained by the GA steered to 20° or 70° from boresight do not have any grating lobes. The highest SLL at 20° is $-6.846\ dB$, (85.3%) suppression compared to the $-1\ dB$ grating lobe. However, the HPBW is still wider but with the grating lobes of the uniform LAA the directivity of all three solutions are superior.

4.4. Chapter Summary

In this chapter we investigated and measured the performance of a uniform LAA spaced 0.5λ apart with uniform amplitude distribution. We then implemented our GA to optimize the amplitude distribution of the signals entering the LAA and compared it to known solutions such Dolph-Chebyshev and Binomial distributions. The GA obtained amplitude distribution with superior SLL reduction compared to the other solutions with only a slight increase in the HPBW compared to the uniform distribution.

The GA was then used to steer a uniform LAA spaced 0.5λ apart with uniform amplitude distribution to different target directions. The results were compared to the steering equation and it was shown that it matched the performance of the steering equation to within 1% accuracy. Proving that the GA could be used to steer the beam of a antenna array to any target direction.

We then implemented the GA optimize the spacing of a LAA when steered to the CA (of the ULA) and beyond and discovered three different antenna spacing that suppressed grating lobes. These solutions out performed the uniform LAA in steering range and SLL reduction but had a wider HPBW.

Chapter 5

Conclusion

5.1. Summary

From our investigation of grating lobes (GL) in a linear antenna array (LAA), it is concluded that the element spacing affects the position of the grating lobes and the size of the main beam's half-power beamwidth (HPBW) and side lobe level (SLL). The phase shift of the signal entering the antennas only influences the position of the main beam while the amplitude distribution affects the SLL and HPBW.

For a uniform, LAA spaced half a wavelength apart, the angle at which the lowest SLL becomes larger than -3 dB which qualifies the sidelobe as a GL is 57° from boresight. This means the steering range of a uniform LAA is 114° degrees of the 180° (or 63.33%) of the visible space. At the critical angle, the HPBW is 36° and the SLL is -3.191 dB .

A genetic algorithm was designed and deployed to determine the best spacing between a 6 antenna LAA to suppress grating lobes. Three solutions were documented in this report that at the mentioned CA has no grating lobes. The highest SLL of one of the solutions at the CA is 7.119 dB below that of the uniform LAA. The HPBW of these solutions ranges between 3.5° to 10° degrees wider than the uniform LAA. This can be undesirable depending on the application. There is a clear trade-off between the SLL and HPBW which would depend on the intended application.

For a narrow HPBW, a wider spacing between antenna elements is required however, this will increase the SLL and decrease the range at which the main beam can be steered because it increases the chance of a GL appearing. For a low SLL, a narrower spaced LAA should be considered but this will increase the HPBW. It is shown that the average spacing between antennas should not be greater than half a wavelength although a single spacing between two antennas can be greater than half a wavelength to ensure grating lobes are suppressed.

It is shown that a non-uniformly spaced LAA suppress the grating lobes of the uniform

LAA when steered to and beyond the critical angle. Depending on the specifications of the intended use the steering range can vary between 110 to 140 degrees (or 61.11% to 77.78%). This means the intended application for an LAA will greatly influence the element spacing and the digital parameters that are needed to steer the main beam.

5.2. Future Work

Our recommendation for future work is to investigate planer antenna arrays. Expanding the LAA to 2-dimensions open up interesting techniques such as density tapering, different rotation designs and sub-arrays that potentially suppress grating lobes. From our understanding of LAA a planer antenna array will be more directive and allow for beam steering in an entire hemisphere.

Bibliography

- [1] I. Rosu. Antenna types and antenna patterns. [Online]. Available: <https://www.qsl.net/va3iul/Antenna/Antenna%20Types%20and%20Antenna%20Patterns.pdf>
- [2] D. K. Cheng, *Field and Wave Electromagnetics*. Pearson, 2014.
- [3] N. R. A. Observatory. The national radio astronomy observatory. [Online]. Available: <https://public.nrao.edu/telescopes/vla/>
- [4] SARAO. Meerkat radio telescope. [Online]. Available: <https://www.sarao.ac.za/gallery/meerkat/>
- [5] S. Magazine. Harris corporation receives funds for phased array radar. [Online]. Available: <https://www.afcea.org/content/Blog-harris-corporation-receives-funds-phased-array-radar>
- [6] D. K. R. Packard. and R. Thomas, *Unequally spaced broadband antenna arrays*. ..., 1960.
- [7] H. Unz, “Linear arrays with arbitrarily distributed elements,” *IRE Transactions on Antennas and Propagation*, vol. 8, no. 2, pp. 222–223, 1960.
- [8] R. Harrington, “Sidelobe reduction by nonuniform element spacing,” *IRE Transactions on Antennas and Propagation*, vol. 9, no. 2, pp. 187–192, 1961.
- [9] M. Ridwan, M. Abdo, and E. Jorswieck, “Design of non-uniform antenna arrays using genetic algorithm,” in *13th International Conference on Advanced Communication Technology (ICACT2011)*, 2011, pp. 422–427.
- [10] J. Khalilpour, J. Ranjbar, and P. Karami, “A novel algorithm in a linear phased array system for side lobe and grating lobe level reduction with large element spacing,” *Analog integrated circuits and signal processing*, vol. 104, no. 3, pp. 265–275, 2020.
- [11] L. S. Matsepe, “Suppression of Grating Lobes,” *Stellenbosch Thesis*, vol. 1, no. 10, pp. 2–28, 2020.
- [12] M. N.M.C., “Broadband configurations for the Square Kilometer Array,” *Eindhoven University of Technology*, pp. 2–28, 2000.
- [13] Z. D. B.P Lathi, *Modern Digital and Analog Communication Systems*. Oxford University Press, 2010.

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

Elo	Chapter(s)	Description
1. Problem solving	3 and 4	The parameters that create and influence grating lobes for a antenna array were investigated and optimized to suppress grating lobes.
2. Application of scientific and engineering knowledge	1, 2	Knowledge of the problem was used to design an algorithm to suppress the grating lobes.
3. Engineering design	3 and 4	I was able to design and implement a genetic algorithm that optimizes the parameters of a LAA to suppress grating lobes while steering the main beam to a desired direction.
4. Investigations, experiments and data analysis	4	Simulated results were obtained, documented, analysed and explained.
5. Engineering methods, skills and tools, including information technology	4	MATLAB's antenna designer toolkit and a genetic algorithm developed in the python programming language was used to solve the problem statement.
6. Professional and technical communication	1, 2, 3, 4, 5	This written report showcases how I can effectively communicate technical information to the reader.
8. Individual work	1, 2, 3, 4, 5	All the research that went into studying and understanding the literature and then applying the knowledge and developing the genetic algorithm was done individually.
9. Independent learning ability	2, 3, 4, 5	All the research that went into studying and understanding the literature and learning how to use the software to develop an algorithm to solve the problem statement was learned individually.

Table B.1: Outcome and Compliance.

Appendix C

More Results

C.0.1. Computer code (python) Genetic Algorithm for optimizing the different parameters of a LAA.

```
1 import numpy as np
2 from functions import *
3 class EvoAmp:
4     #Initiate parameters
5     def __init__(self, population_tot, crossover_rate, mutation_rate, generations, N, d, target):
6         self.population_tot = population_tot
7         self.crossover_rate = crossover_rate
8         self.mutation_rate = mutation_rate
9         self.generations = generations
10        self.N = N
11        self.target = target
12
13    # Spacing between antennas
14    self.pop_dis = np.ndarray(shape=(self.population_tot, self.N-1))
15
16    # Phase Shift of the signal of each antenna element
17    self.pop_pha = np.random.uniform(size = (self.population_tot, self.N), low = 0, high = 2*np.pi*0) # radians
18
19    # Amplitude of the signal of each antenna element.
20    self.pop_amp = np.random.uniform(size=(self.population_tot, self.N), low = 0, high=1)
21
22    # Population with same spacing and phase -> only optimizing amplitude
23    for i in range(self.population_tot):
24        self.pop_dis[i, :] = d
25
26    # Fitness scores
27    self.pop_fit = np.zeros(self.population_tot, dtype = np.float)
28
29    self.evolve()
30
31    def results(self):
32        # Return the best solution
33        q = np.argmax(self.pop_fit)
34        return self.pop_amp[q]
35
36    def evolve(self):
37        # probabilities for cross over. To avoid calculationg random numbers every itteration.
38        crossovers = np.random.uniform(size=(self.generations), low = 0, high=1)
39        mutations = np.random.uniform(size=(self.generations), low = 0, high=1)
40
41        for i in range(self.generations):
42            # Calculate the fitness score of each solution
43            for dna in range(self.population_tot):
44                score = fitness(self.pop_dis[dna], self.pop_pha[dna], self.pop_amp[dna], self.target)
45                self.pop_fit[dna] = score;
46            # Index of fittest solution
47            index = np.argmax(self.pop_fit)
48            # Copy fittest solution over to next generation (first and last index of array)
49            self.pop_amp[0] = self.pop_amp[index]
50            self.pop_amp[-1] = self.pop_amp[index]
51
52            # Stop simulating when the generation matches the maximun no of generations.
53            if i == self.generations-1:
54                print('DONE')
55                break
56            # Iterations per generations
57            for _ in range(1):
58
59                if crossovers[i] < self.crossover_rate:
60                    # crossover occurs (Two parent solutions)
61                    s_1 = selection(self.pop_fit)
62                    s_2 = selection(self.pop_fit)
63                    self.pop_amp[s_1] = (self.pop_amp[s_1] + self.pop_amp[s_2])/2
64
65                if mutations[i] < self.mutation_rate:
66                    # mutation occurs
67                    s_1 = selection(self.pop_fit)
68                    self.pop_amp[s_1][np.random.randint(low=0, high=self.N)] = np.random.uniform(low = 0, high=1)
```

Figure C.1: GA that optimizes the amplitude of the signal entering a LAA of size N to any desired direction.

```

1 import numpy as np
2 from evolution import selection, fitness
3 from functions import *
4
5 class EvoPhase:
6     #Initiate parameters
7     def __init__(self, population_tot, crossover_rate, mutation_rate, generations, N, d, target):
8         self.population_tot = population_tot
9         self.crossover_rate = crossover_rate
10        self.mutation_rate = mutation_rate
11        self.generations = generations
12        self.N = N
13        self.target = target
14
15    # Spacing between antennas
16    self.pop_dis = np.ndarray(shape=(self.population_tot, self.N-1))
17
18    # Phase Shift of the signal of each antenna element
19    self.pop_pha = np.random.uniform(size = (self.population_tot, self.N), low = 0, high = 2*np.pi) # radians
20
21    # Amplitude of the signal of each antenna element.
22    self.pop_amp = np.ndarray(shape=(self.population_tot, self.N))
23    # Uniform amplitude
24    amp = np.array([1, 1, 1, 1, 1, 1])
25
26    # Population with same spacing and amplitude -> Only optimizing phase
27    for i in range(self.population_tot):
28        self.pop_dis[i, :] = d
29        self.pop_amp[i, :] = amp
30
31    # Fitness scores
32    self.pop_fit = np.zeros(self.population_tot, dtype = np.float)
33
34    self.evolve()
35
36    def results(self):
37        # Return the best solution
38        q = np.argmax(self.pop_fit)
39        return self.pop_pha[q]
40
41    def evolve(self):
42        # probabilities for crossover. To avoid calculating random numbers every iteration.
43        crossovers = np.random.uniform(size=(self.generations), low = 0, high=1)
44        mutations = np.random.uniform(size=(self.generations), low = 0, high=1)
45
46        for i in range(self.generations):
47            tot = 0
48            # Calculate the fitness score of each solution
49            for dna in range(self.population_tot):
50                score = fitness(self.pop_dis[dna], self.pop_pha[dna], self.pop_amp[dna], self.target)
51                self.pop_fit[dna] = score;
52                tot += score
53            # Index of fittest solution
54            index = np.argmax(self.pop_fit)
55            # Copy fittest solution over to next generation (first and last index of array)
56            self.pop_pha[0] = self.pop_pha[index]
57            self.pop_pha[-1] = self.pop_pha[index]
58
59            # Stop simulating when the generation matches the maximum no of generations.
60            if i == self.generations-1:
61                print('DONE')
62                break
63
64            # Iterations per generations
65            for _ in range(1):
66                if crossovers[_] < self.crossover_rate:
67                    # crossover occurs (Two parent solutions)
68                    s_1 = selection(self.pop_fit)
69                    s_2 = selection(self.pop_fit)
70                    # Offspring
71                    self.pop_pha[s_1] = np.divide(self.pop_pha[s_1] + self.pop_pha[s_2],2)
72
73                if mutations[_] < self.mutation_rate:
74                    # mutation occurs
75                    s_1 = selection(self.pop_fit)
76                    self.pop_pha[s_1][np.random.randint(low=0, high=self.N)] = np.random.uniform(low = 0, high=2*np.pi)
77

```

Figure C.2: GA that only optimizes the phase for a LAA of size N to any desired direction.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from functions import *
4 from evophase import *
5
6 global LOW, HIGH
7 LOW = 0.3
8 HIGH = 0.6
9
10 class Evolution:
11     #Initiate parameters
12     def __init__(self, population_tot, crossover_rate, mutation_rate, generations, N, target):
13         self.population_tot = population_tot
14         self.crossover_rate = crossover_rate
15         self.mutation_rate = mutation_rate
16         self.generations = generations
17         self.N = N
18         self.target = target
19
20     # Spacing between antennas
21     self.pop_dis = np.ndarray(shape=(self.population_tot, self.N-1))
22
23     # Non-uniform symmetrically spaced LAA
24     for i in range(self.population_tot):
25         a = np.random.uniform(low = LOW, high=HIGH)
26         b = np.random.uniform(low = LOW, high=HIGH)
27         self.pop_dis[i][4] = a;
28         self.pop_dis[i][0] = a
29         self.pop_dis[i][1] = b
30         self.pop_dis[i][3] = b
31
32         self.pop_dis[i][2] = np.random.uniform(low = LOW, high=HIGH)
33
34     # Phase shift of the signal of each antenna element.
35     self.pop_pha = np.random.uniform(size = (self.population_tot, self.N), low = 0, high = 2*np.pi) # radians
36
37     # Amplitude of the signal of each antenna element.
38     self.pop_amp = np.random.uniform(size=(self.population_tot, self.N), low = 1, high=1)
39
40     # Fitness scores
41     self.pop_fit = np.zeros(self.population_tot, dtype = np.float)
42
43     self.evolve()
44
45     def results(self):
46         # Return the best solution
47         q = np.argmax(self.pop_fit)
48         return self.pop_dis[q], self.pop_pha[q], self.pop_amp[q]
49
50     def evolve(self):
51         # Probabilities for crossover. To avoid calculating random numbers every iteration.
52         crossovers = np.random.uniform(size=(self.generations), low = 0, high=1)
53         mutations = np.random.uniform(size=(self.generations), low = 0, high=1)
54         # Probabilities for which parameter to mutate or to undergo crossover
55         choice = np.random.uniform(size=(self.generations), low = 0, high=1)
56
57         for i in range(self.generations):
58
59             for dna in range(self.population_tot):
60                 score = fitness(self.pop_dis[dna], self.pop_pha[dna], self.pop_amp[dna], self.target)
61                 self.pop_fit[dna] = score;
62             # Index of fittest solution
63             index = np.argmax(self.pop_fit)
64             # Copy fittest solution over to next generation (first and last index of array)
65             self.pop_dis[0] = self.pop_dis[index]
66             self.pop_pha[0] = self.pop_pha[index]
67             self.pop_dis[-1] = self.pop_dis[index]
68             self.pop_pha[-1] = self.pop_pha[index]
69
70             # Optimize the phase shift for the fittest solution.
71             evo = EvoPhase(5, 0.5, 1, 2000, self.N, self.pop_dis[0], self.target)
72             self.pop_pha[0] = evo.results()
73
74             # Stop simulating when the generation matches the maximum no of generations.
75             if i == self.generations-1:
76                 print('Finished')
77                 break
78
79             # Iterations per generations
80             for _ in range(1):
81                 if crossovers[_] < self.crossover_rate:
82                     # crossover occurs (Two parent solutions)
83                     s_1 = selection(self.pop_fit)
84                     s_2 = selection(self.pop_fit)
85
86                     # Only share spacing DNA
87                     if choice[_] <= 0.5:
88                         self.pop_dis[s_1] = (self.pop_dis[s_1] + self.pop_dis[s_2]) / 2
89                     # Only share phase DNA
90                     elif choice[_] > 0.5:
91                         self.pop_pha[s_1] = (self.pop_pha[s_1] + self.pop_pha[s_2]) / 2
92
93                     if mutations[_] < self.mutation_rate:
94                         # mutation occurs
95                         s_1 = selection(self.pop_fit)
96                         # Only mutate spacing DNA
97                         if choice[_] <= 0.5:
98                             rs = np.random.uniform(low = LOW, high=HIGH)
99                             a = np.random.randint(low=0, high=self.N-1)
100                            # SYMMETRICAL SPACING
101                            self.pop_dis[s_1][a] = rs
102                            self.pop_dis[s_1][(self.N-2 - a)] = rs
103                            # Only mutate phase DNA
104                            elif choice[_] > 0.5:
105                                self.pop_pha[s_1][np.random.randint(low=0, high=self.N)] = np.random.uniform(low = 0, high=2*np.pi)

```

Figure C.3: GA that optimizes the antenna element spacing and the phase shift for a LAA of size N to any desired direction.