



UNIVERSIDAD DE BUENOS AIRES

MASTER THESIS ON ELECTRONIC ENGINEERING

Computational design of slot waveguide antennas

Author:
Gonzalo Becker

Directors:
Dr. Leonardo Rey Vega
Dr. Christophe Calvez
Ing. Franck Houllier

*Department of Electronics
Faculty of Engineering*

June 27, 2025

Acknowledgements

At the completion of this thesis, I wish to express my deepest gratitude to all those who made this work possible:

To my advisors, for their expert guidance, insightful feedback, and unwavering support throughout this journey. Your deep knowledge of RF design and meticulous reviews have been invaluable to the success of this research.

To my professors, whose passion for science and dedication to teaching sparked my curiosity and drove me to explore new frontiers in the use of innovative computational methods for the design of electronic devices

To my family, for being my rock at every step: thank you for your love, patience, and constant encouragement. Without your understanding and motivation, this endeavor would not have been possible.

To my friends and colleagues, for your camaraderie, stimulating discussions, and daily support.

Contents

1	Introduction	13
1.1	Motivation	13
1.2	Objectives	13
1.3	Tools and Skills	14
1.4	Approaches to SWA Design	14
1.5	Tools	15
2	Antenna theory	17
2.1	Antenna characterization	17
2.1.1	Directivity	17
2.1.2	Gain	17
2.1.3	Side-Lobe Level (SLL)	18
2.1.4	Polarization	18
2.1.5	Impedance Matching	18
2.1.6	Efficiency	18
2.2	Antenna Arrays	18
2.2.1	Linear Arrays	19
2.2.2	Chebyshev distribution	20
2.3	3-dimensional radiation patterns	21
2.4	Coordinate Systems	22
3	Radar Systems	25
4	Theoretical foundations of SWAs	27
4.1	Wave propagation in waveguides	27
4.2	Transmission Line Analogy	32
4.3	Design equations	34
4.4	Coupling corrections	35
5	Resonant slot optimization	37
5.1	Antenna simulation considerations	37
5.1.1	Ground plane dimensions estimation	37

5.1.2	Waveguide length computation	38
5.1.3	Microstrip-to-waveguide transition	39
5.2	Single slot characterisation	40
5.2.1	Variation of the offset and length	40
5.2.2	Single Slot Radiation Pattern	45
5.2.3	Slot width variation	45
5.2.4	Waveguide wall width variation	45
5.2.5	Slot characterisation for non ideal case	45
5.3	Antenna design	47
5.3.1	Slot voltages	47
5.3.2	Ideal case	47
5.3.3	Non ideal case	50
5.4	Cross and co-Polarization	52
5.5	Optimization on the 3D Radiation Pattern	53
5.5.1	Optimization based on out of FoV power	54
5.5.2	Optimization based on a target radiation pattern	56
5.5.3	Azimuth Plane beam shaping	59
6	Non-resonant Slot SWA Optimization	61
6.1	Theoretical Model	61
6.1.1	ABCD Parameters	61
6.1.2	Wave Propagation in a Transmission Line with Shunt Elements	62
6.1.3	Waveguide Voltage Calculation	62
6.1.4	Slot admittance characterization	62
6.1.5	Design Equations	63
6.2	Antenna solver Algorithm	63
6.2.1	Self-Consistent Solution Algorithm	64
6.3	Genetic Algorithm for Slot Array Optimization	68
6.3.1	Geometry Constraint Processing	68
6.3.2	Genetic Algorithm Components	68
6.3.3	Voltage Inheritance Acceleration	70
6.3.4	Population Initialization Strategy	70
6.3.5	Integration with Antenna Solver	70
6.3.6	Parallelization Strategy	72
6.4	Results	73
6.4.1	Hyperparameter choice	73
6.4.2	Algorithm evolution	73
6.4.3	Best solution	73
7	Metasurface Optimisation	75
7.1	HFSS Scripting	75
7.1.1	Grid approach	75
7.1.2	Vectorized approach	76

7.2	Random-walk optimization	77
7.2.1	Results	79
7.3	HFSS Scripting	79
7.3.1	Grid approach	80
7.3.2	Vectorized approach	80
7.4	Random-walk optimization	81
7.5	Random Antenna Generation	84
7.5.1	Stochastic region generation	84
7.5.2	Region growth mechanism	85
7.6	Discrete Metasurface Graph Neural Network	85
7.7	Graph Definitions	86
7.8	KNN-based pooling	87
7.9	Message Passing Neural Networks (MPNNs)	88
7.10	Auto-Encoder as a geometric descriptor	89
7.10.1	Encoder	90
7.10.2	Decoder	91
7.10.3	Training Objective	92
7.10.4	Hyperparameter choice	93
7.10.5	Results	93
7.11	Architecture	95
7.11.1	Regularization	95
7.11.2	Positional Encoding	96
7.12	SWA Metasurface Theoretical Model	96
7.12.1	Slot Voltage Prediction	96
7.12.2	Radiation Pattern Predictions	97
7.13	Earth Moving Distance (EMD) for radiation patterns	97
7.13.1	Motivation	97
7.13.2	Definition	98
7.13.3	Sinkhorn approximation	98
7.13.4	GeomLoss - hyperparameter choice	98
7.14	Training	99
7.14.1	Overfitting Test	99
7.14.2	Hyperparameter choice	102
7.14.3	Results	103
8	Conclusion	109

List of Figures

2.1	Linear antenna array configuration showing N radiating elements with spacing d	19
2.2	Uniform amplitude distribution for 5 colinear radiating elements	19
2.3	Chebyshev amplitude distribution with controlled side lobe levels	19
2.4	Chebyshev amplitude coefficients for optimal array design	20
2.5	Chebyshev polynomials of different orders showing their characteristic properties	20
2.6	Spatial disposition of radiating elements showing positions and offsets	21
2.7	3D radiation pattern for 5 ideal colinear radiating elements	22
2.8	3D radiation pattern for offseted radiating elements (SWA configuration)	22
2.9	Comparison of 3D radiation patterns between ideal and practical SWA configurations	22
2.10	Single longitudinal slot radiation pattern obtained from electromagnetic simulation	23
2.11	Comparison of different coordinate system transformations used in antenna measurements and simulations	23
2.12	Ludwig2 coordinate system used in antenna measurement benchmarks	24
4.1	Paired views of the electromagnetic field for each TE_{mn} mode. Left: cross-sectional (yz) "transversal" fields. Right: longitudinal (xz) "top-view" field norms.	31
4.2	Surface currents in the top face of a rectangular waveguide operating in TE_{10} mode, showing the distribution that guides slot placement optimization	32
4.3	SWA geometry and transmission line analogy showing slot placement at resonant positions. Note that the open circuit (closed waveguide) is on the left side	33
4.4	Geometric configuration for computing coupling coefficients between slots, showing distances R , R_1 , and R_2 used in the mutual coupling calculations	35
5.1	Single slot radiation pattern analysis for different ground plane dimensions to determine minimum required size	38
5.2	Azimuth plane radiation pattern comparison for different ground plane scaling factors p , showing convergence at $p = 6$	38
5.3	Waveguide length correction factor δ determination through parametric analysis, showing optimal value of $33.83\mu m$	39
5.4	Transition design	40
5.5	S	40

5.6	Single slot waveguide schematic showing measurement setup and equivalent circuit parameters for admittance characterization	41
5.7	Simulated conductance (G) vs slot length for different offsets (ideal case)	42
5.8	Simulated susceptance (B) vs slot length for different offsets (ideal case)	42
5.9	Resonant length $l_r(x)$ normalized to free space wavelength as a function of slot offset	43
5.10	Normalized conductance $\hat{G}(\hat{l})$ showing universal behavior across different offsets	44
5.11	Normalized susceptance $\hat{B}(\hat{l})$ showing universal behavior across different offsets	44
5.12	Spline interpolation of normalized conductance and susceptance for universal slot behavior modeling	44
5.13	Comparison between theoretical Elliott's equation and simulated resonant conductance values	44
5.14	Single slot elevation radiation pattern for offset = 1.5 mm and length = 2.5 mm (resonant length)	45
5.15	Simulated conductance (G) vs slot length for different offsets (non-ideal case: h=0.5mm, w=0.5mm)	46
5.16	Simulated susceptance (B) vs slot length for different offsets (non-ideal case: h=0.5mm, w=0.5mm)	46
5.17	Resonant length variation with offset for non-ideal slot geometry	46
5.18	Simulated Susceptance (B)	46
5.19	Theoretical array factor for Chebyshev distribution with optimized side lobe level for 20° field of view	47
5.20	Elevation radiation pattern results for ideal 5-slot SWA with Chebyshev distribution	48
5.22	Enter Caption	49
5.21	Azimuth radiation pattern results for ideal 5-slot SWA with Chebyshev distribution	49
5.23	Enter Caption	50
5.24	Enter Caption	51
5.25	Enter Caption	51
5.26	Enter Caption	52
5.27	Enter Caption	52
5.28	Enter Caption	53
5.29	Enter Caption	54
5.30	out of FoV radiation pattern	55
5.31	Comparison of initial vs. final field of view outputs.	55
5.32	Enter Caption	56
5.33	Target 3D radiation pattern optimized for radar specifications	57
5.34	2D slices showing -3dB and -10dB specifications verification	57
5.35	Target 3D radiation pattern (left) and -3dB and -10dB specifications verification (right)	57
5.36	Absolute difference between target radiation pattern and estimated radiation pattern	58

5.37 Convergence behavior of the target pattern optimization algorithm showing improvement over iterations	59
6.1 Slot admittance values as a function of offset and length	63
6.2 Antenna solving algorithm flowchart	67
6.3 Integration flowchart showing genetic algorithm interaction with antenna solver. Blue boxes represent GA operations, green boxes represent solver operations, yellow boxes represent decision points.	71
6.4 Evolution of the GA	73
6.5 Slot disposition	74
6.6 Slot amplitudes and phases	74
7.1 Comparison of the grid and vectorized simulations	76
7.2 Random-walk optimization algorithm flow diagram	77
7.3 Material-distribution maps: (a) initial solution, (b) optimized solution (500 iterations)	79
7.4 Evolution of the optimization probability map: (a) initial, (b) after finding best solution, (c) mid-optimization random iteration.	79
7.5 Final optimized antenna structure as simulated in ANSYS HFSS	80
7.6 Random-walk optimization algorithm flow diagram	82
7.7 Random antenna generation algorithm flow	84
7.8 2-stage KNN graph pooling with a ration 5:1/ Same regions are represented by the same color.	87
7.9 Auto-Encoder architecture to extract geometric descriptors	90
7.10 Encoder detailed Architecture	91
7.11 Encoder detailed Architecture	92
7.12 Enter Caption	94
7.13 Enter Caption	94
7.14 Enter Caption	94
7.15 Enter Caption	95
7.16 Enter Caption	95
7.17 Enter Caption	96
7.18 Enter Caption	97
7.19 Top: input Yin. Middle: co-prediction vs ground truth. Bottom: cross-prediction vs ground truth.	100
7.20 Training and validation losses	103
7.21 Comparison of decomposed training versus validation losses	103
7.22 Loss Histograms	104
7.23 Detailed Loss histograms	104
7.24 Top: input Yin. Middle: co-prediction vs ground truth. Bottom: cross-prediction vs ground truth.	105
7.25 Loss Histograms	106
7.26 Loss Histograms	106

7.27 Loss Histograms	107
7.28 Loss Histograms	107
7.29 Comparison of decomposed training versus validation losses	108
7.30 Comparison of decomposed training versus validation losses	108

Chapter 1

Introduction

The rapid development of high-frequency wireless systems, particularly in automotive radar and next-generation communication networks, has renewed interest in Slotted Waveguide Antennas (SWAs). SWAs leverage precisely engineered apertures on metallic waveguides to achieve high radiation efficiency, mechanical robustness, and controlled beam characteristics. Traditionally designed via extensive electromagnetic (EM) simulations, SWAs demand significant computational resources when optimizing slot parameters and array configurations. This thesis proposes a unified computational framework that merges analytical models and modern optimization techniques to accelerate SWA design while preserving or enhancing performance metrics.

1.1 Motivation

As autonomous vehicles and 5G/6G networks proliferate, system architects require compact, low-cost radar and antenna solutions capable of operating reliably under diverse conditions. SWAs offer an attractive balance between efficiency and manufacturability, yet classical design methods, such as those based on Stevenson's equivalence model or Elliott's mutual coupling algorithm, often yield suboptimal results when limited to few slots or nonideal waveguide configurations. Furthermore, emerging antenna paradigms, including metasurface-based designs and machine-learning-driven layouts, suggest new avenues to circumvent limitations of traditional approaches. By exploring a spectrum of design strategies—from first-principles algorithms to data-driven optimization—this work aims to deliver a flexible, high-fidelity toolkit for SWA synthesis across varied application scenarios.

1.2 Objectives

The primary goal of this thesis is the development and validation of a computational design methodology for SWAs that:

- Reduces reliance on costly EM simulations through hybrid analytical-numerical models.
- Integrates multiple optimization paradigms, including Elliott's mutual coupling algorithm, metasurface antenna techniques, and neural-network-based optimization.
- Demonstrates robustness for both high-slot-count arrays and minimal-slot (e.g., five-slot) configurations relevant for compact radar sensors.
- Validates designs against performance criteria: reflection coefficient ($S_{11} < -15$ dB over 76–77 GHz), directivity, side-lobe level ratio (SLR > 20 dB), and beamwidth specifications (azimuth $\pm 25^\circ$, elevation $\pm 9^\circ$).

1.3 Tools and Skills

This thesis employs a multidisciplinary toolset:

EM Simulation: Ansys HFSS for high-fidelity modelling of waveguide structures and slot apertures.

Computational Platforms: Python for data processing, scripting of simulation workflows, and implementation of analytical algorithms.

Optimization Frameworks: SciPy and custom Python modules for gradient-based and evolutionary optimization, alongside PyTorch for neural approaches.

Visualization: Matplotlib and Plotly for radiation pattern and parameter visualization.

1.4 Approaches to SWA Design

Elliott's design Algorithm

Building upon Stevenson's slot-equivalence model, Elliott's algorithm accounts for near-field coupling between slots to improve amplitude and phase predictions. This work implements Elliott's formulation in Python, automates slot admittance extraction via HFSS simulations, and compares results against analytical benchmarks for arrays up to 20 slots.

Metasurface Antenna Optimization

Metasurface antennas, comprising subwavelength patterned surfaces, enable fine-grained control of wavefronts beyond classical slot arrays. By discretizing the waveguide sidewall as a metasurface, this thesis explores gradient-based optimization of slot shapes and placements, exploiting metasurface theory to tailor local boundary conditions and achieve prescribed radiation patterns with fewer elements.

Neural-Network-Based Optimization

Recent advances in machine learning facilitate rapid antenna design by learning complex EM behaviors. A surrogate model, trained on a database of HFSS simulations, predicts key performance metrics (e.g., S_{11} , directivity) as functions of slot parameters. This surrogate accelerates global search with evolutionary algorithms or reinforcement learning, enabling exploration of novel slot geometries and arrangements.

Through the integration of these methodologies—analytical, metasurface-based, and neural—the thesis aims to establish a comprehensive and efficient framework for the next generation of SWA design.

1.5 Tools

Ansys HFSS has the possibility of performing optimisations based on different approaches.

Chapter 2

Antenna theory

In this section we study different elements of antennas and antenna arrays. Firstly, an insight onto radiation patterns, polarisation, power and gain considerations and matching levels of antennas is given. Then, the theoretical frameworks for modeling antenna arrays from amplitude distributions over the radiating elements is established. Later in this work, the objective will be to generate amplitude distributions of an array according to given specifications and, conversely, to predict the radiation pattern from a given distribution.

2.1 Antenna characterization

This section introduces key figures of merit used to evaluate antenna performance: directivity, gain, side-lobe level (SLL), polarization, impedance matching, phase matching, and efficiency.

2.1.1 Directivity

Directivity D measures the concentration of radiated power in a given direction relative to an isotropic radiator:

$$D(\theta, \phi) = \frac{U(\theta, \phi)}{U_{\text{iso}}} = \frac{4\pi U(\theta, \phi)}{P_{\text{rad}}} \quad (2.1)$$

where $U(\theta, \phi)$ is the radiation intensity (W/sr) and P_{rad} is the total radiated power.

2.1.2 Gain

Gain G accounts for antenna efficiency η in addition to directivity:

$$G(\theta, \phi) = \eta D(\theta, \phi), \quad 0 < \eta \leq 1. \quad (2.2)$$

Here, $\eta = P_{\text{rad}}/P_{\text{input}}$ captures losses in conductors and dielectrics.

2.1.3 Side-Lobe Level (SLL)

The Side-Lobe Level quantifies the relative power of the highest secondary lobe compared to the main lobe, typically expressed in decibels:

$$\text{SLL} = 20 \log_{10} \frac{P_{\text{SL}}}{P_{\text{ML}}} \quad [\text{dB}], \quad (2.3)$$

where P_{SL} is the side-lobe power and P_{ML} is the main-lobe power.

2.1.4 Polarization

Polarization describes the orientation of the electric field vector. For linear polarization:

$$\mathbf{E}(t) = E_{0x} \hat{x}, e^{j(\omega t + \phi_x)} + E_{0y} \hat{y}, e^{j(\omega t + \phi_y)}. \quad (2.4)$$

Circular polarization occurs when $E_{0x} = E_{0y}$ and $\phi_x - \phi_y = \pm 90^\circ$.

2.1.5 Impedance Matching

Good matching minimizes reflections. The reflection coefficient is

$$\Gamma = \frac{Z_{\text{in}} - Z_0}{Z_{\text{in}} + Z_0}, \quad (2.5)$$

and the Voltage Standing Wave Ratio (VSWR) is

$$\text{VSWR} = \frac{1 + |\Gamma|}{1 - |\Gamma|}. \quad (2.6)$$

2.1.6 Efficiency

Radiation efficiency is defined as

$$\eta = \frac{P_{\text{rad}}}{P_{\text{input}}} = 1 - |\Gamma|^2 - \alpha_{\text{loss}}, \quad (2.7)$$

where α_{loss} accounts for ohmic and dielectric losses.

2.2 Antenna Arrays

Antenna arrays can be of many types, such as linear, circular or planar. As a first approximation, it will be considered that the radiating slots in a SWA define a linear array.

The array factor (AF) of an antenna array is supposed to convey all of the information contained on a disposition of radiating elements. For instance, a single element, with a classical isotropic radiation intensity given as a function of distance r by $e^{-jkr}/(4\pi r)$, the AF is considered to be $AF(\theta) = 1$.

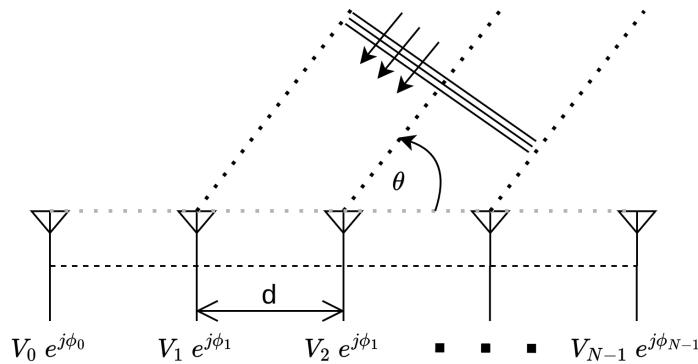


Figure 2.1: Linear antenna array configuration showing N radiating elements with spacing d

2.2.1 Linear Arrays

It can be proven that the phase shift between consecutive antennas is $\Delta\phi = \phi_{i+1} - \phi_i = kd \cos(\theta)$. Here, θ is the angle of an incident (or emitted) planar waveform, and $k = 2\pi/\lambda_0$ is the wavenumber. The array factor is defined by

$$AF = \sum_{m=0}^{N-1} V_m e^{j\phi_m} = \sum_{m=0}^{N-1} V_m e^{jm \Delta\phi} \quad (2.8)$$

If $V_m = 1 \forall m$, it is said that the amplitude distribution is uniform, in which case the squared Array Factor is displayed in Figure 2.2. Here, the $-3dB$ and $-10dB$ levels define the field of view (FoV). The main disadvantages of a uniform distribution is that no control is given over the relationship between the FoV and the side lobe level (SLL).

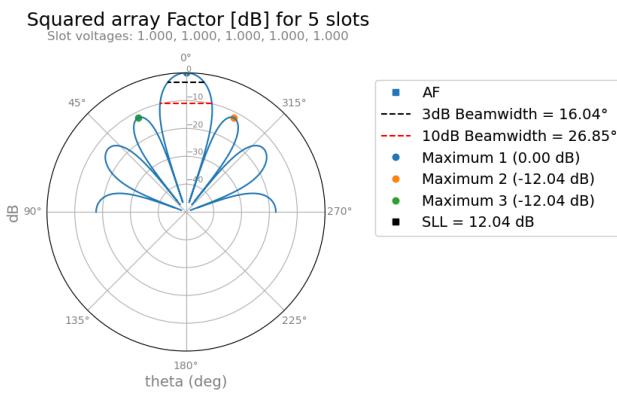


Figure 2.2: Uniform amplitude distribution for 5 colinear radiating elements

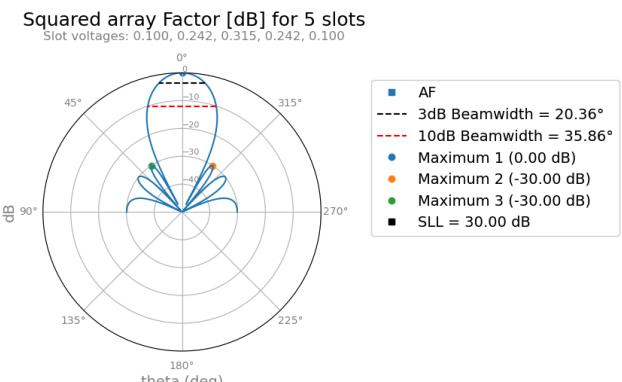


Figure 2.3: Chebyshev amplitude distribution with controlled side lobe levels

2.2.2 Chebyshev distribution

When distributions are not constant, a better control of the radiation diagram is achieved. This is the case in Figure 2.3, where a Chebyshev distribution is shown.

To obtain these coefficients, a theoretical background on Chebyshev polynomials is needed. Chebyshev polynomials are polynomials functions defined by the recursive relationship $T_m(z) = 2zT_{m-1}(z) - T_{m-2}(z)$. They are depicted in Figure 2.5. As clearly seen, they have the property of having all roots in the interval $(-1, 1)$, with all of their local maximum and minimum values in that interval being ± 1 .

At the same time, it is possible to prove that the array factor can be expressed as a sum of cosine functions, due to the complex exponential properties. For example, for an array with an even number of elements $2M$ and symmetric amplitude coefficients $V_n = V_{2M-n-1}$, the array will have an array factor given by

$$AF = \sum_{n=1}^M V_n \cos((2n-1)u) \quad (2.9)$$

where $u = \pi d \cos(\theta)/\lambda_0$. At the same time, there is an identity by which

$$\cos(mu) = T_m(\cos(u)) \quad (2.10)$$

This last relationship means that the array factor can be expressed as a sum of Chebyshev polynomials. To exploit this property, it will be searched that the array factor is identical to a Chebyshev polynomial of order one less than the total number of elements in the array.

For this purpose, as shown in Figure 2.4, a value $z_0 > 1$ is computed such that $T_{2M-1}(z_0) = R_0$, where R_0 is the ratio between the searched peak and the first side lobes (that is, R_0 in dB represents the side lobe ratio). This polynomial, in the range $(0, z_0)$, will be defined as the objective array factor. For this purpose, a variable change is performed, $\cos(u) = z/z_0$, which leads to the relationship

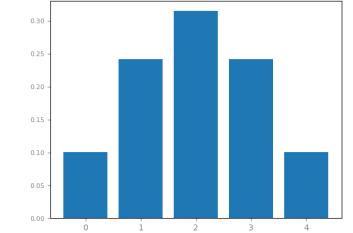


Figure 2.4: Chebyshev amplitude coefficients for optimal array design

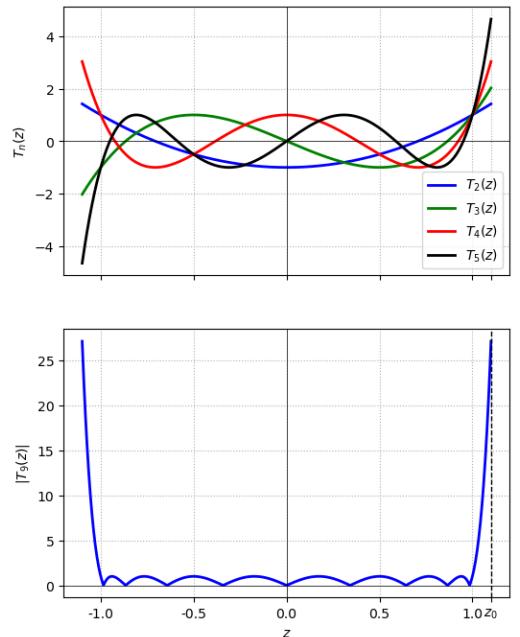


Figure 2.5: Chebyshev polynomials of different orders showing their characteristic properties

$$AF = \sum_{n=1}^{2M} V_n T_{2n-1} \left(\frac{z}{z_0} \right) = T_{2M-1}(z) \quad (2.11)$$

By equating the polynomials the coefficients V_n can be obtained.

There exists an interesting compromise relationship between the FoV, the gain and the side lobe level, since, if the size of the antenna is to be kept constant, for narrower FoVs it is required to redirect more power towards the side lobes, decreasing therefore the effectiveness of the antenna.

2.3 3-dimensional radiation patterns

Let's suppose, from a theoretical stand, that there are N colinear radiating elements of intensity V_i along the x axis.

We proceed to establish the theory of antenna arrays. Notably, the array factor (AF) is the resulting radiation pattern for a given disposition of radiating elements in space. Given slot voltage $V[n]$ and equally spaced colinear radiating elements along the x axis, separated by a distance d , the following formula holds:

$$AF(\theta) = \sum_{n=1}^N V[n] \cdot e^{\frac{2\pi j (n - \frac{N-1}{2}) \cdot d \cdot \cos(\theta)}{\lambda_0}} \quad (2.12)$$

The total radiated electric field is a function of θ and can be computed by the following relationship:

$$\mathbf{E}(\text{total}) = [\mathbf{E}(\text{element unitaire})] \times [AF(\theta)] \quad (2.13)$$

Using these formulas, the total radiation pattern can be plotted for 2 array distributions, one where radiating offsets are completely colinear, and another where they are offset as in a SWA. The results are shown in Figure 2.9.

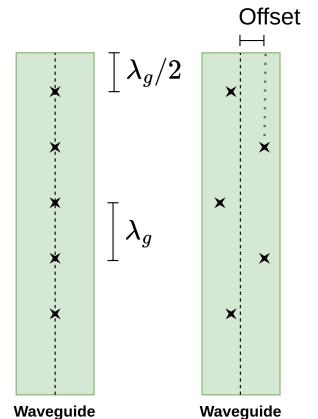


Figure 2.6: Spatial disposition of radiating elements showing positions and offsets

3D Radiation Pattern

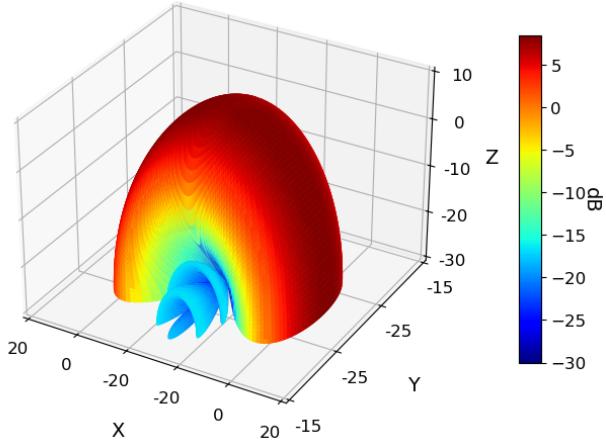


Figure 2.7: 3D radiation pattern for 5 ideal colinear radiating elements

3D Radiation Pattern

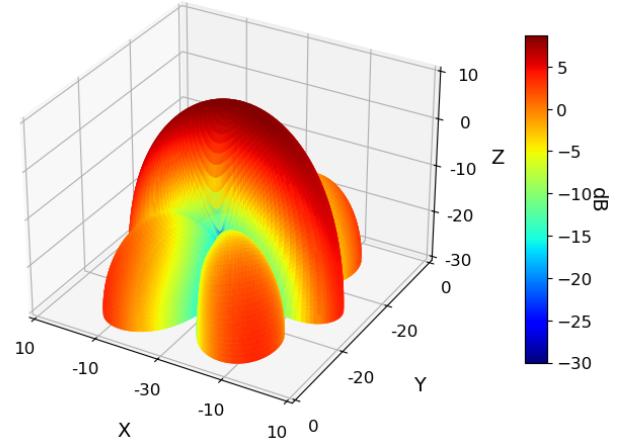


Figure 2.8: 3D radiation pattern for offsetted radiating elements (SWA configuration)

Figure 2.9: Comparison of 3D radiation patterns between ideal and practical SWA configurations

The single longitudinal slot radiation pattern is obtained from simulation. By multiplying this 3D radiation pattern, element by element, with the array factor, a good approximation for the total radiation pattern should be obtained. The single slot radiation pattern is shown in Figure 2.10.

2.4 Coordinate Systems

In what follows, conversion formulas between the standard Cartesian coordinate system and other coordinate systems often used in radar systems will be given. Refer to Figure 2.11 to understand each coordinate system. In all cases, the blue arrow is the vector to be represented and has an amplitude ρ and Cartesian coordinates (x, y, z) .

The spherical coordinate system is the most classically used system in physics. It is the coordinate system in which results are usually exported from HFSS software. Unless otherwise stated, this is the coordinate system that will be used throughout this work. The conversion formulas to the Cartesian coordinate system are given by:

$$\begin{aligned} x &= \rho \sin(\theta) \cos(\varphi) \\ y &= \rho \sin(\theta) \sin(\varphi) \\ z &= \rho \cos(\theta) \end{aligned} \tag{2.14}$$

Single slot Gain

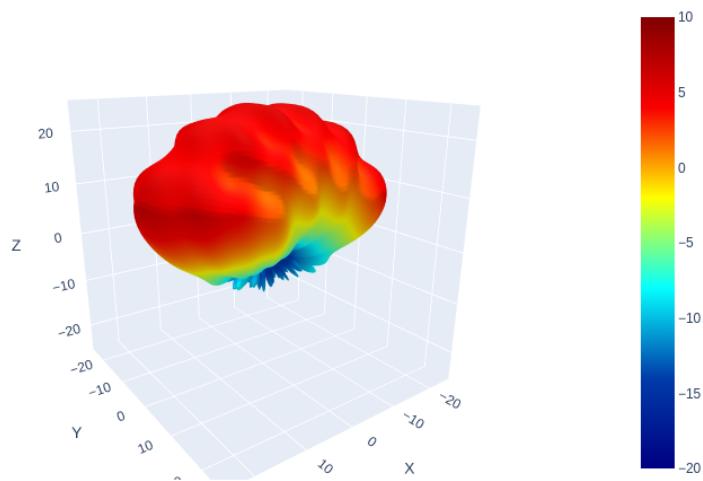
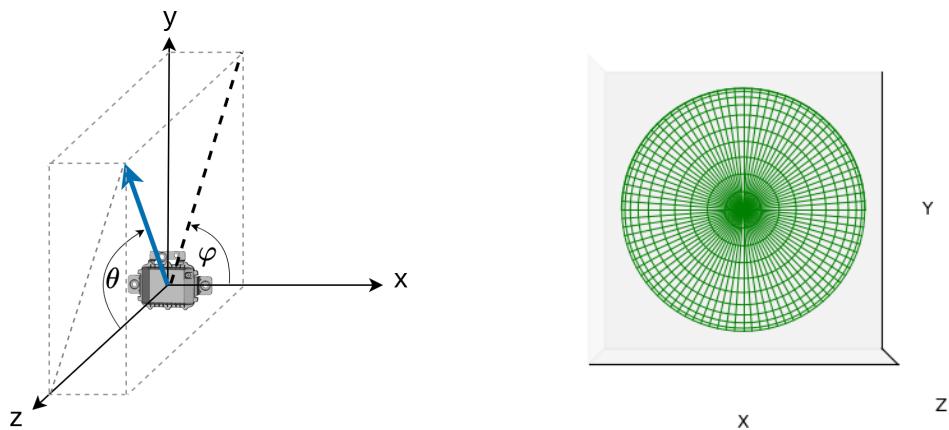
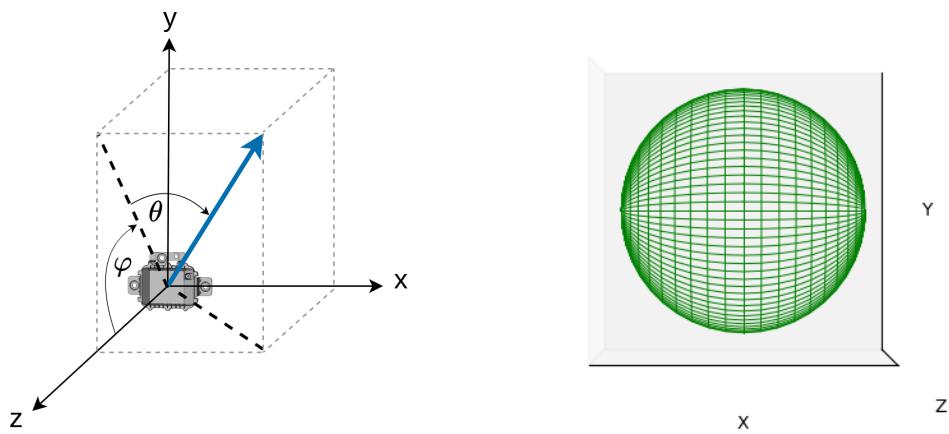


Figure 2.10: Single longitudinal slot radiation pattern obtained from electromagnetic simulation

Spherical coordinates representation**Ludwig2 coordinates representation**

The ludwig2 coordinate system is theFor the ludwig2 Elevation over Azimuth coordinate system, it is as follows:

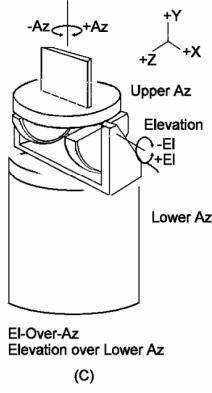


Figure 2.12: Ludwig2 coordinate system used in antenna measurement benchmarks

The Ludwig2 coordinate system is the coordinate representation used in several antenna characterisation benchmarks. As depicted in Figure 2.12, the mechanical restrictions of many measurement setups requires that first a rotation is performed over one angle, and after a rotation is performed on the other rotation angle. It is also possible to export simulation data from CST software in this coordinate system.

Its conversion formulas to the cartesian system are:

$$\begin{aligned} x &= \rho \sin(\theta) \\ y &= \rho \cos(\theta) \sin(\varphi) \\ z &= \rho \cos(\theta) \cos(\varphi) \end{aligned} \quad (2.15)$$

Finally, as can be seen in Figure 2.11, the alternative system has two rotation angles on the xz and yz planes, leading to a rotation sphere with 4 vertices. This system has the advantage of clearly allowing to specify the Fields of View (FoV) for both Elevation and Azimuth projections. For example, by choosing a range of θ , $(-\theta_{3dB}, \theta_{3dB})$, it is clear that a FoV has been defined for the vertical cut $y = 0$. Moreover, this coordinate system will prove to be particularly useful when trying to define a target 3D radiation diagram (XXX). The conversion formulas are given below.

$$\begin{aligned} x &= r \cdot \sin(\tan(\theta) \cdot \cos(\varphi)) \\ y &= r \cdot \sin(\tan(\cos(\theta) \cdot \tan(\varphi))) \\ z &= \sqrt{1 - \sin^2(\tan(\tan(\theta) \cdot \cos(\varphi))) - \sin^2(\tan(\cos(\theta) \cdot \tan(\varphi)))} \end{aligned} \quad (2.16)$$

It is of particular interest, for later applications, to find conversion formulas from the spherical coordinate system to the alternative system. Let θ_s and ϕ_s be the spherical angles, and θ_a and ϕ_a the alternative ones.

When analysing the alternative system, it becomes clear that

$$\frac{x}{z} = \tan(\theta_a) \quad \frac{y}{z} = \tan(\phi_a) \quad (2.17)$$

By replacing x , y , z by the formulas in 2.14, the following result is obtained:

$$\begin{aligned} \theta_a &= \text{atan}(\tan(\theta_s) \cos(\phi_s)) \\ \phi_a &= \text{atan}(\tan(\theta_s) \sin(\phi_s)) \end{aligned} \quad (2.18)$$

Chapter 3

Radar Systems

Chapter 4

Theoretical foundations of SWAs

SWAs have been studied for decades, leading today to a deep knowledge about their operation. In this section, their theoretical explanation will be given, and the algorithm proposed by Elliot and Stevenson will be detailed.

4.1 Wave propagation in waveguides

In order to obtain the propagation relations for the waveguide, free-space Maxwell equations must be stated:

$$\nabla \cdot \mathbf{E} = 0 \quad (4.1)$$

$$\nabla \cdot \mathbf{H} = 0 \quad (4.2)$$

$$\nabla \times \mathbf{E} = -j\omega\mu\mathbf{H} \quad (4.3)$$

$$\nabla \times \mathbf{H} = j\omega\epsilon\mathbf{E} \quad (4.4)$$

Inside a waveguide, propagation takes places in the (\hat{z}) direction, so that it is possible to establish the following equations:

$$\mathbf{E}(x, y, z, t) = \mathbf{E}(x, y)e^{j(\omega t - \beta z)} \quad (4.5)$$

$$\mathbf{H}(x, y, z, t) = \mathbf{H}(x, y)e^{j(\omega t - \beta z)} \quad (4.6)$$

β is known as the wave number and $\lambda_g = 2\pi/\beta$ is the guide wavelength.

In order to solve these equations, fields \mathbf{E} and \mathbf{H} are decomposed in their tangential and orthogonal components, and we therefore analyse separately the equations for each of these directions. This allows to obtain the following results:

$$\nabla_T E_z \times \hat{\mathbf{z}} - j\beta \hat{\mathbf{z}} \times \mathbf{E}_T = -j\omega\mu \mathbf{H}_T \quad (4.7)$$

$$\nabla_T H_z \times \hat{\mathbf{z}} - j\beta \hat{\mathbf{z}} \times \mathbf{H}_T = j\omega\mu \mathbf{E}_T \quad (4.8)$$

$$\nabla_T \times \mathbf{E}_{\mathbf{T}} + j\omega\mu \hat{\mathbf{z}} H_z = 0 \quad (4.9)$$

$$\nabla_T \times \mathbf{H}_{\mathbf{T}} - j\omega\epsilon \hat{\mathbf{z}} E_z = 0 \quad (4.10)$$

$$\nabla_T \cdot \mathbf{E}_T - j\beta E_z = 0 \quad (4.11)$$

$$\nabla_T \cdot \mathbf{H}_T - j\beta H_z = 0 \quad (4.12)$$

By using the first two equations, the following relation is obtained:

$$\mathbf{E}_T = -\frac{j\beta}{k_c^2} (\nabla_T E_z - \eta_{TE} \hat{\mathbf{z}} \times \nabla_{\mathbf{T}} H_z) \quad (4.13)$$

$$\mathbf{H}_T = -\frac{j\beta}{k_c^2} (\nabla_T H_z + \frac{1}{\eta_{TM}} \hat{\mathbf{z}} \times \nabla_{\mathbf{T}} E_z) \quad (4.14)$$

Where the following variables have been defined:

Transversal impedance of TE mode:

$$\eta_{TE} = \frac{\omega\mu}{\beta} \quad (4.15)$$

Transversal impedance of TM mode:

$$\eta_{TM} = \frac{\beta}{\omega\epsilon} \quad (4.16)$$

Cut-off wave number:

$$k_c^2 = \omega^2\epsilon\mu - \beta^2 = k^2 - \beta^2 \quad (4.17)$$

Cut-off frequency:

$$\omega_c = c k_c \quad (4.18)$$

Cut-off Wavelength:

$$\lambda_c = \frac{2\pi}{k_c} \quad (4.19)$$

Guided wave length:

$$\lambda_g = \frac{\lambda}{\sqrt{1 - \left(\frac{\lambda}{\lambda_c}\right)^2}} \quad (4.20)$$

By taking the last four equations previously obtained, we get now Helmholtz equations:

$$\nabla_T^2 E_z + k_c^2 E_z = 0 \quad (4.21)$$

$$\nabla_T^2 H_z + k_c^2 H_z = 0 \quad (4.22)$$

For a waveguide operating in the TE mode ($E_z = 0$), the found equations become:

$$\nabla_T^2 H_z + k_c^2 H_z = 0 \quad (4.23)$$

$$\mathbf{H}_T = -\frac{j\beta}{k_c^2} \nabla_T H_z \quad (4.24)$$

$$\mathbf{E}_T = \eta_T E \mathbf{H}_T \times \hat{z} \quad (4.25)$$

We then search for solutions that can be factorized as $H_z(x, y) = F(x)G(y)$, so that Helmholtz equations now become:

$$F''(x)G(y) + F(x)G''(y) + k_c^2 F(x)G(y) = 0 \Rightarrow \frac{F''(x)}{F(x)} + \frac{G''(y)}{G(y)} + k_c^2 = 0$$

From where we realise that the terms depending on F and G must be constant. By solving the differential equations, we find the fields for TEM_{mn} modes:

$$\begin{aligned} H_z(x, y) &= H_1 \cos(k_x x) \cos(k_y y) \\ H_x(x, y) &= H_1 \sin(k_x x) \cos(k_y y) \end{aligned}$$

$$H_y(x, y) = H_2 \cos(k_x x) \sin(k_y y)$$

$$E_z = 0$$

$$E_x(x, y) = E_1 \cos(k_x x) \sin(k_y y)$$

$$E_y(x, y) = E_2 \sin(k_x x) \cos(k_y y)$$

with constants

$$H_1 = \frac{j\beta k_x}{k_c^2} H_0$$

$$H_2 = \frac{j\beta k_y}{k_c^2} H_0$$

$$E_1 = \eta_{TE} H_2 = j\eta \frac{\omega k_y}{\omega_c k_c} H_0$$

$$E_2 = -\eta_{TE} H_1 = -j\eta \frac{\omega k_x}{\omega_c k_c} H_0$$

and with

$$k_x = \frac{n\pi}{a}$$

$$k_y = \frac{m\pi}{b}$$

$$k_c = \sqrt{\left(\frac{n\pi}{a}\right)^2 + \left(\frac{m\pi}{b}\right)^2}$$

All fields are multiplied by $e^{j(\omega t - \beta z)}$

These equations then allow to plot in python the transversal electromagnetic fields in the waveguide for different propagation modes, as can be seen in image

Also, for a frequency that only allows the TE₀₁ mode to propagate, that is $f_{cut-off01} < \min(f_{cut-off10}, f_{cut-off11})$, the intensities of the electric and magnetic fields can be plotted from a top view of the waveguide (xz plane) (See Figure ??)

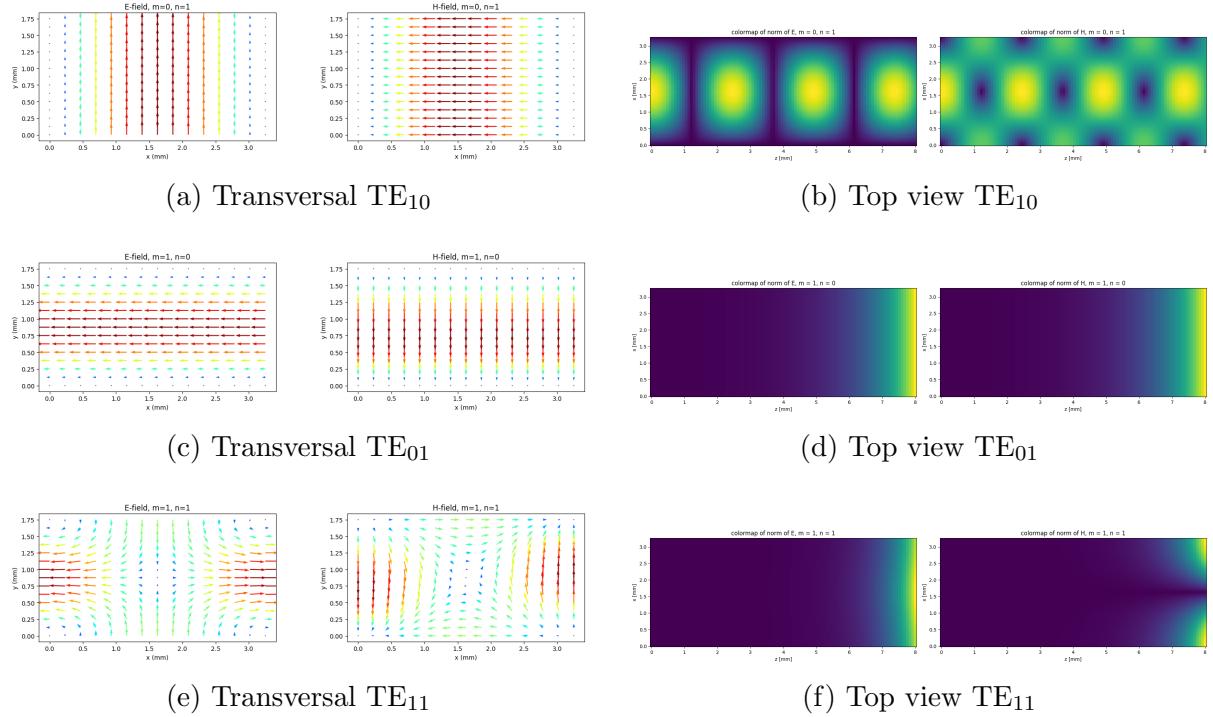


Figure 4.1: Paired views of the electromagnetic field for each TE_{mn} mode. Left: cross-sectional (yz) "transversal" fields. Right: longitudinal (xz) "top-view" field norms.

We can then calculate surface currents:

$$\mathbf{J}_s = \hat{z} \times (\mathbf{H}_1 - \mathbf{H}_2) \quad (4.26)$$

where $H_2 = 0$ because it is the external field.

We finally obtain the surface currents, which result useful to understand the SWA operation.

$$\mathbf{J}_s = -H_z(x, y)\hat{x} + H_x(x, y)\hat{z} \quad (4.27)$$

It is therefore clear that wave guides allow for a resonant behaviour of electromagnetic waves inside them. In Figure 4.2, the surface currents in the waveguide top face have been plotted, and they allow to understand how should slots be placed. Firstly, it is clear that, since polarization must be linear and horizontal (considering that the wave guide will be vertical), then the simplest solution is to place longitudinal slots. Then, it is proven that slots will tend to irradiate more if the perpendicular surface currents they cut are stronger. For instance, given that surface currents are stronger near the edges of the top face, then radiation intensity, and therefore equivalent amplitude, will be stronger as we get towards the edges. Finally, slots must be placed in an alternating manner, because the surface currents show a phase-induced variation throughout the wave guide and so alternating positions are necessary to ensure that the electric field is constructive in the main axe of the antenna. Of course this defines an

approximative intuition of its behavior, since, once slots are placed, the current distribution must necessarily change.

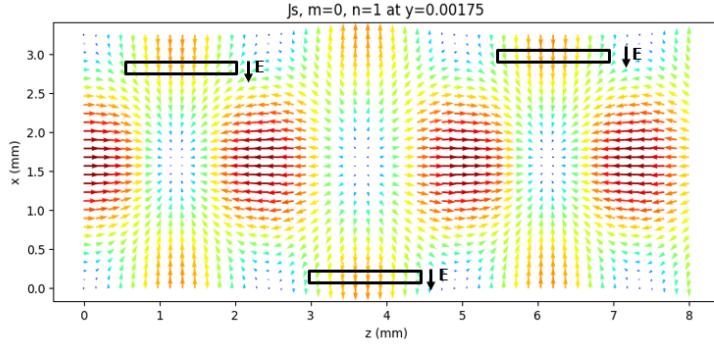


Figure 4.2: Surface currents in the top face of a rectangular waveguide operating in TE_{10} mode, showing the distribution that guides slot placement optimization

4.2 Transmission Line Analogy

Regarding the geometry, the waveguide is defined by its guided wave length (λ_g). By placing slots in resonant positions (separated by $\lambda_g/2$) we ensure that all irradiating elements (the slots) will be in phase. However, the last slot is placed at a distance $\lambda_g/4$ of the open circuit at the end of the waveguide.

In a simple fashion, an approximation could be made to say that each slot, while radiating energy to the surroundings, is behaving as an impedance, or admittance, for the antenna. In Figure 5.5, it can be observed the geometry of the antenna as well as the transmission line analogy, where each slot is modelled by an admittance.

Regarding the geometry, slots are positioned at resonant positions in the waveguide, so that all slots will irradiate with the same phase. The end of the waveguide is closed, what, from a transmission line perspective, can be thought of as an open circuit (reflexion coefficient equal to 1, since all of the signal is reflected).

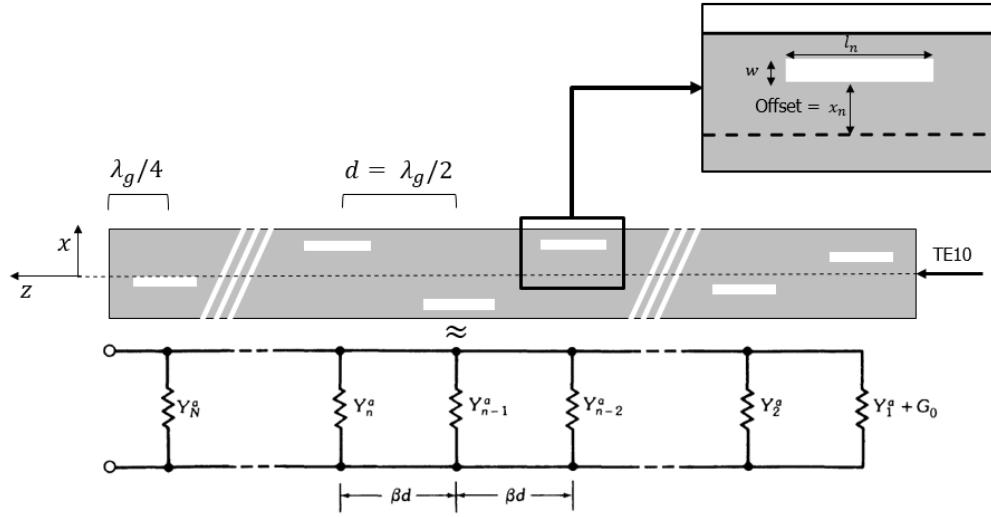


Figure 4.3: SWA geometry and transmission line analogy showing slot placement at resonant positions. Note that the open circuit (closed waveguide) is on the left side

While designing the antenna, a specific matching level of the system waveguide - antenna will be required. Usually, a complete match will be required. In that case, the reflection coefficient will be given by

$$\rho = \frac{Z_L - Z_0}{Z_L + Z_0} \quad (4.28)$$

if $\rho = 0$ is required, then $Z_L = Z_0$, which finally implies that the grouped effect of all the slots must give an equivalent impedance of Z_0 . Here, Z_0 is the characteristic impedance of the waveguide. Switching to admittance, and under the supposition that the waveguide is a lossless transmission line, we get $Y_0 = 1/Z_0 = G_0 + jB_0 = G_0$, where G_0 is the conductance and B_0 the susceptance.

In the context of transmission line theory, it can be proven that, given a load Z_L seen from a distance L in a transmission line, the observed input impedance is

$$Z_{in}(-L) = Z_0 \frac{Z_L + jZ_0 \tan(\beta L)}{Z_0 + jZ_L \tan(\beta L)} \quad (4.29)$$

by replacing $\beta = 2\pi/\lambda_g$ and $L = \lambda_g/2$, the terms containing $\tan(\beta L) = \tan(\pi) = 0$ go to 0, so we get $Z_{in}(-L) = Z_L$. This implies that if all slots are separated by $L = \lambda_g/2$, then all of their equivalent admittances could be thought of as being connected to a same node, so that their combined total equivalent admittance is then

$$Y_{input} = \sum_{i=1}^n Y_i \quad (4.30)$$

If G_0 is the characteristic admittance, we then seek

$$Y_{input} = \sum_{i=1}^n Y_i = G_0 \Rightarrow \sum_{i=1}^n \frac{Y_i}{G_0} = 1 \quad (4.31)$$

4.3 Design equations

Once the transmission line analogy established, it is now imperative to establish the design equations, first established by Stevenson and Elliot, among others, in order to model the relationship between equivalent admittances and the amplitude distributions as functions of the geometry of each slot.

Firstly, a relation is established linking the active admittance (called active because it considers coupling effects) of each slot, the offset, the length, its amplitude distribution V^s and the common mode voltage, V_n related to the propagation in the waveguide.

$$\frac{Y_n^a}{G_0} = K_1 f(x_n, l_n) \frac{V_n^s}{V_n} \quad (4.32)$$

Here,

$$K_1 = \frac{2(\pi/a)^2}{j\omega\mu_0(\beta_{10}/k)(ka)(kb)} \quad (4.33)$$

$$f(x, l) = \frac{(\pi/2kl)\cos(\beta_{10}l)}{(\pi/2kl)^2 - (\beta_{10}/k)^2} \sin\left(\frac{\pi x}{a}\right) \quad (4.34)$$

$$V^s = \int_{x_0-w/2}^{x_0+w/2} E_x(x', 0) dx' \quad (4.35)$$

Then, the fact that all slots are placed in resonant positions implies that $V_n = V_m \forall m, n$, so that, for any two nth and mth slots, the following relation is obtained:

$$\frac{Y_n^a/G_0}{Y_m^a/G_0} = \frac{|f(x_n, l_n)|V_n^s}{|f(x_m, l_m)|V_m^s} \quad (4.36)$$

4.4 Coupling corrections

In Figure 4.4 the diagram to compute the coupling coefficients is observed. By this means, the idea is that the distances R , R_1 and R_2 are functions of the positions z_n and z_m of the nth and mth slot respectively. This will define a coupling coefficient matrix, where each term reflects the complex coefficient g_{mn} that allows to correct for the near-fields coupling between the 2 slots.

This relationship is given by

$$g_{mn} = \int_{-k_0 l_m}^{k_0 l_m} \cos\left(\frac{u'_m}{4l_m/\lambda_0}\right) \left(\frac{1}{4l_n/\lambda_0} \left(\frac{e^{-jk_0 R_1}}{k_0 R_1} + \frac{e^{-jk_0 R_2}}{k_0 R_2} \right) \right. \\ \left. + \left(1 - \frac{1}{(4l_n/\lambda_0)^2}\right) \int_{-k_0 l_n}^{k_0 l_n} \cos\left(\frac{u'_n}{4l_n/\lambda_0}\right) \frac{e^{-jk_0 R}}{k_0 R} du'_n \right) du'_m \quad (4.37)$$

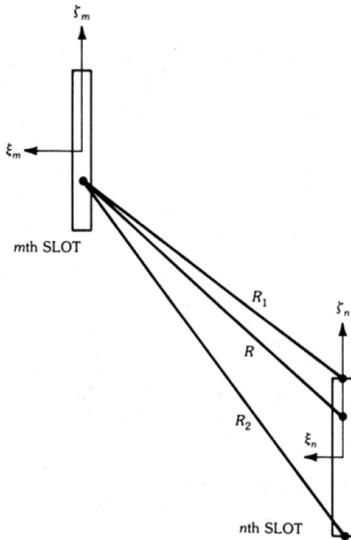


Figure 4.4: Geometric configuration for computing coupling coefficients between slots, showing distances R , R_1 , and R_2 used in the mutual coupling calculations

These coupling coefficients are then used to obtain a final relationship to compute the active admittance of each slot given the coupling coefficients, the offset, the length and the amplitude distribution.

$$\frac{Y_n^a}{G_0} = \frac{2f^2(x_n, l_n)}{\frac{2f^2(x_n, l_n)}{Y(x_n, l_n)/G_0} + j \frac{\beta_{10}}{k} k_0 b \left(\frac{a}{\lambda}\right)^3 \sum_{m=1}^n \frac{V_m^s}{V_n^s} g_{mn}} \quad (4.38)$$

Chapter 5

Resonant slot optimization

In this section, the main objective is to show the results obtained through the implementation of the algorithm described in the previous chapter by using simulations in ANSYS HFSS and processing the data in python.

The different aspects to take into consideration to correctly apply the previous algorithm, including details when generating the 3D model, will be detailed in this section.

Initially, it is intended to validate Elliott's algorithm for a perfect theoretical model. For this purpose, a waveguide of dimensions $a = 3.5\text{mm}$ and $b = 1.75\text{mm}$ will be used, with a negligible wall width $h = 10\mu\text{m}$ and a negligible slot width $w = 50\mu\text{m}$. The reason for choosing $a = 2b$ is to maximize the maximum admissible power in the waveguide before dielectric breakdown takes place. The negligible values of h and w seek to approximate as much as possible the ideal conditions specified in the design algorithm.

Subsequently, other cases where h and w are not ideal will be considered as well. The impact of these parameters over our model will be considered.

5.1 Antenna simulation considerations

When defining the antenna dimensions and building the 3D geometry, there exist some considerations to take into account, which will be explained in this section.

5.1.1 Ground plane dimensions estimation

The theoretical background is established under the supposition that the ground plane over the waveguide is infinite. To approximate the ground plane dimension correctly, an analysis must be carried out to establish which are the minimum ground plane dimensions W_{ground} and L_{ground} to ensure the validity of the implemented algorithm.

By defining a single slot with variable dimensions $W_{ground} = L_{ground} = p\lambda_0$, it is possible to determine the minimum ground plane dimension by increasing p until the radiation pattern does not change significantly.

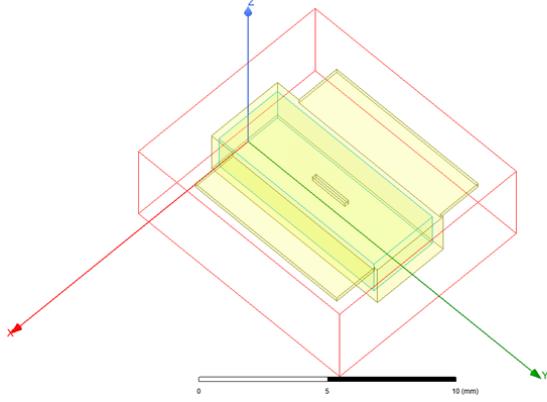


Figure 5.1: Single slot radiation pattern analysis for different ground plane dimensions to determine minimum required size

By varying p , we finally observe that between $p = 6$ and $p = 8$ the difference between the radiation patterns in the azimuth planes is significantly reduced. Since in modern radars many antennas are placed next to each other, it is clear that the theoretical model will not be perfect, so it is possible to choose $p = 6$ without affecting importantly the performance of the algorithm.

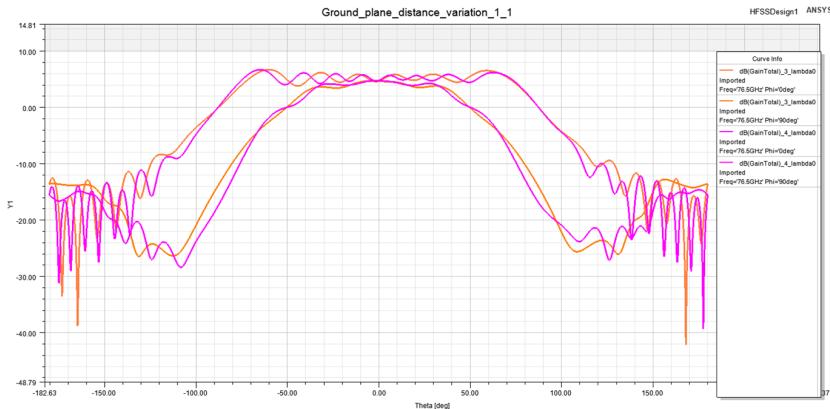


Figure 5.2: Azimuth plane radiation pattern comparison for different ground plane scaling factors p , showing convergence at $p = 6$

5.1.2 Waveguide length computation

To define the waveguide length, one would normally choose it as $\lambda_g (N_{slots}/2)$. However, since the open end is not perfectly matched, there is a correction factor δ that must be added to the length so as to ensure that resonance will be achieved in the desired positions.

As will be seen later on, when doing the simulations for a single slot, the wave port will be placed at a distance of $1.25\lambda_g + \delta$ from the end of the waveguide. By applying the transmission line model from Eq. (4.29), it will later be shown that with this geometry and by placing the slot at a distance of $\lambda_g/2$ from the port, the total admittance seen at the port (both real and imaginary parts) will be equal to $Y_{slot}(x, l)$. If then $x = 0$, which implies that there is practically no slot present in the waveguide, it is expected that $Y_{slot}(0, l) = 0 + 0j$, because with no offset there shouldn't be any radiation. The correction δ must be chosen so that this last condition is verified.

By doing a parametric simulation over δ , the plot in Figure 5.3 is obtained. It is clear then that the correction to be made for this particular case is $\delta = 33.83\mu m$.

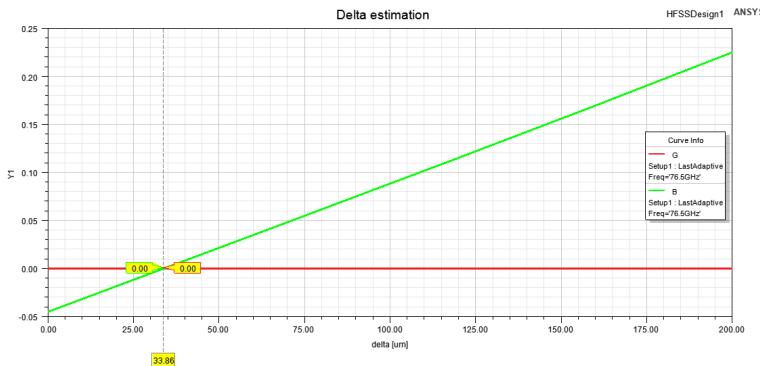


Figure 5.3: Waveguide length correction factor δ determination through parametric analysis, showing optimal value of $33.83\mu m$

5.1.3 Microstrip-to-waveguide transition

While we have assumed perfect matching from the input port of the antenna, in reality there exists a transition from the microstrip line that connects to an embedded system, and the waveguide. This transition can further affect the response of the antenna array.

For instance, as a side-quest of the present work a novel microstrip-to-waveguide transition based on a patch antenna and on stacked PCB layers was designed by using a Genetic algorithm. The design is represented in Figure ??, while S parameters are in Figure 5.5, showing that both S_{11} and S_{22} are negligible at the resonant frequency, and S_{12} is almost 1, as expected for a correct optimisation.

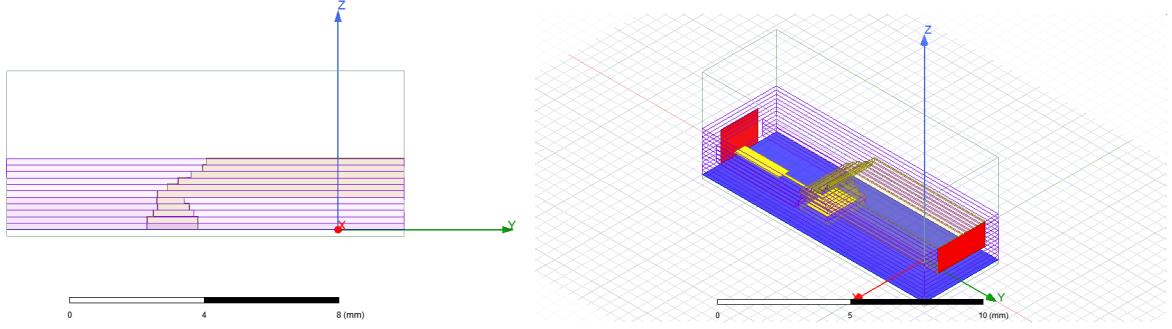


Figure 5.4: Transition design

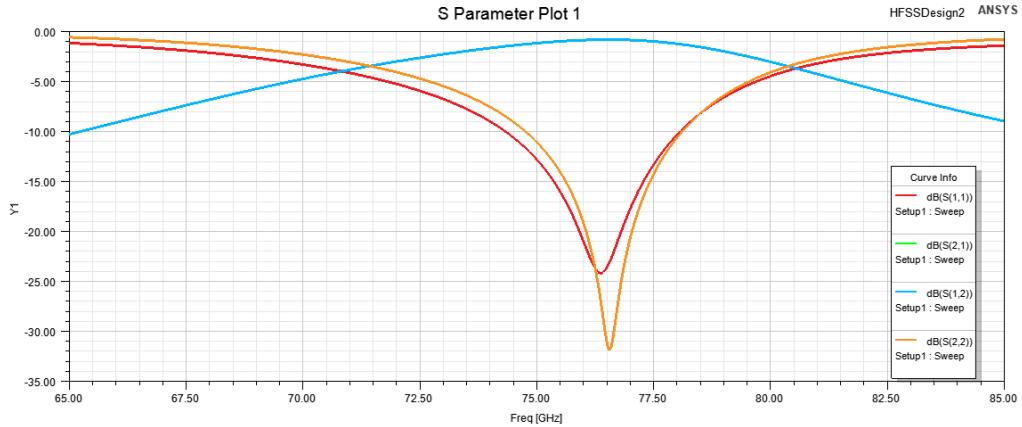


Figure 5.5: S

5.2 Single slot characterisation

5.2.1 Variation of the offset and length

Given the geometry of Figure 5.6, the input impedance seen at the waveport is

$$Y_{in}(-1.25\lambda_g - \delta) = Y_{slot}(x, l) \quad (5.1)$$

To prove this result, we will assume that we are working with a completely rectangular waveguide where $\delta = 0$. Then, by using the transmission line model from Eq. (4.29), we observe that, if the load at $L = 0$ mm is $Z_L = Z_{short} = 0$, then

$$\lim_{L \rightarrow 0.75\lambda_g} \tan(\beta L) = \infty \quad (5.2)$$

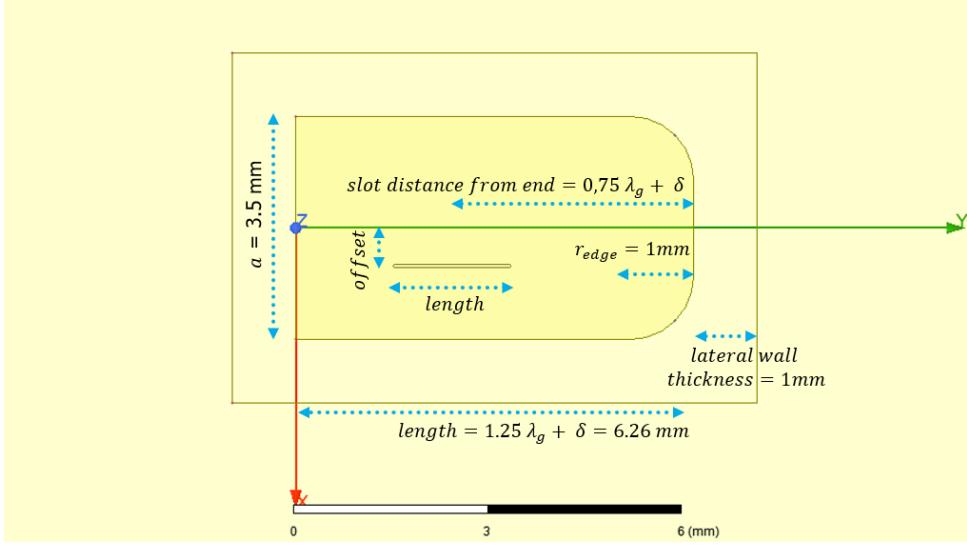


Figure 5.6: Single slot waveguide schematic showing measurement setup and equivalent circuit parameters for admittance characterization

$$Z_{in}(-0.75\lambda_g) = \lim_{L \rightarrow -0.75\lambda_g} Z_0 \frac{Z_L + jZ_0 \tan(\beta L)}{Z_0 + jZ_L \tan(\beta L)} = \infty \quad (5.3)$$

Then, the equivalent load at point $L = -0.75\lambda_g$ is

$$Z'_L = Z_{slot} \parallel Z_{in}(-0.75\lambda_g) = Z_{slot} \quad (5.4)$$

Finally, when $L = -0.5\lambda_g$,

$$\tan(\beta L) = 0 \quad (5.5)$$

so

$$Z_{in}(-1.25\lambda_g) = Z'_{in}(-0.5\lambda_g) = Z_0 \frac{Z_{slot} + jZ_0 \tan(\beta L)}{Z_0 + jZ_{slot} \tan(\beta L)} = Z_{slot} \quad (5.6)$$

from where Eq. (5.1) is induced.

Once this is defined, the objective will be to characterize the slot for all possible pairs of offsets and lengths. For this purpose, a parametric simulation will be carried out in HFSS, and the equivalent slot admittance will be computed for each simulation as

$$Y_{slot} = Y_{in} = \frac{1 - S_{11}}{1 + S_{11}} \quad (5.7)$$

The obtained results are shown in Figure 5.7 and Figure 5.8:

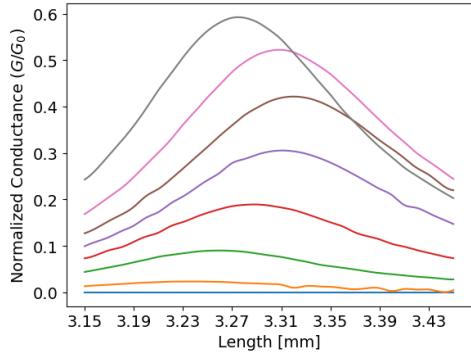


Figure 5.7: Simulated conductance (G) vs slot length for different offsets (ideal case)

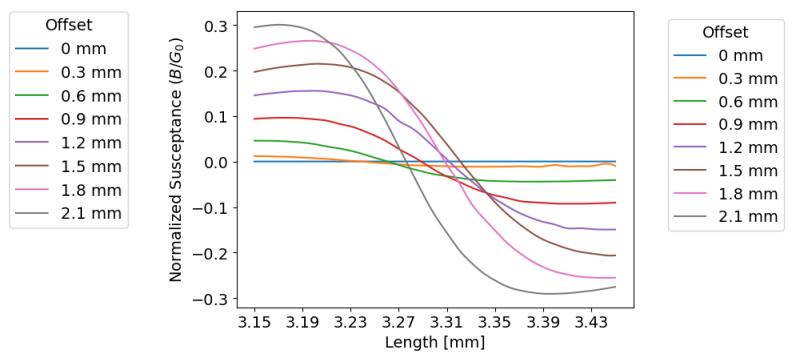


Figure 5.8: Simulated susceptance (B) vs slot length for different offsets (ideal case)

It is interesting to observe that, in those points where conductance G is maximum, susceptance B takes a value very close to 0. This allows us to define a length $l_r(x)$ where

$$Y(x, l_r(x)) = G_r(x) + j \cdot 0 \quad \forall x \quad (5.8)$$

$l_r(x)$ will be defined as the resonant length of a slot with an offset x , because this length maximizes conductance, minimizes susceptance and therefore maximizes radiation amplitude. $G_r(x)$ will then denote the maximum value of conductance for each offset x .

In Figure 5.9 we observe the plot of $l_r(x)$ normalized to free space wavelength against x . For low offsets, it is reasonable to expect an unstable behavior, since both conductance and susceptance take low values. It is also interesting to note that the bibliography usually considers the resonant length to be about $0.5\lambda_0$, although it is clear that in practice resonance is achieved at a lower length.

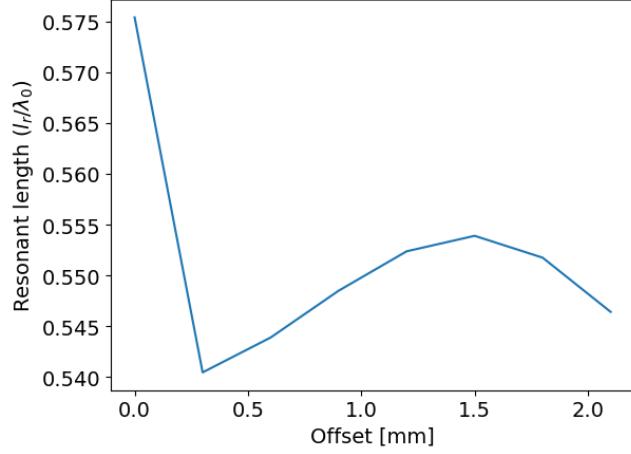


Figure 5.9: Resonant length $l_r(x)$ normalized to free space wavelength as a function of slot offset

At this point, given the similarity in the shape of different lines for different values of offset, a normalization of the different lines will be searched. For this purpose, both conductance and susceptance values will be normalized by the resonant conductance (maximum of G), and the length x axis will be normalized by the resonant length, for each offset.

In other words, for each offset x_0 and by defining $\hat{l} = l/l_r(x_0)$ the following lines will be plotted:

$$\hat{G}(\hat{l}) = \frac{G(x_0, \hat{l} l_r(x_0))}{G_r(x_0)}, \quad \hat{B}(\hat{l}) = \frac{B(x_0, \hat{l} l_r(x_0))}{G_r(x_0)} \quad (5.9)$$

More intuitively, it is a change of variables that allows to rescale all lines with respect to their maximum value and maximum's position.

In Figures 5.10 and 5.11, both of these lines are plotted. Small offsets are erased for stability reasons.

A spline interpolation is carried out to obtain the results shown in Figure 5.12:

In [?], Elliott proposed Eq. (5.10) to estimate the conductance of a slot as a function of its offset:

$$\frac{G(x, l)}{G_0} = \frac{73}{R_d} \left(\frac{4(a/b)}{0.61\pi(\beta/k)} (\cos(\beta l) - \cos(kl))^2 \sin^2 \left(\frac{\pi x}{a} \right) \right) \quad (5.10)$$

In particular, this equation can be used to predict theoretically $G_r(x) = G(x, l_r(x))$. The comparison between the simulated results and the theoretical equations is shown in Figure 5.13.

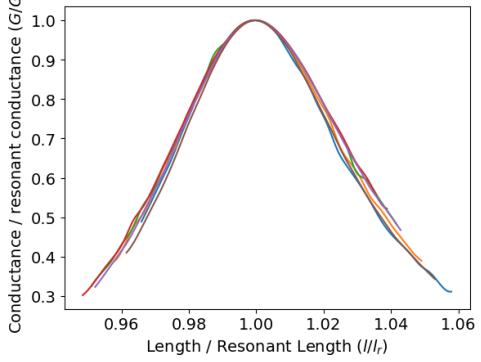


Figure 5.10: Normalized conductance $\hat{G}(\hat{l})$ showing universal behavior across different offsets

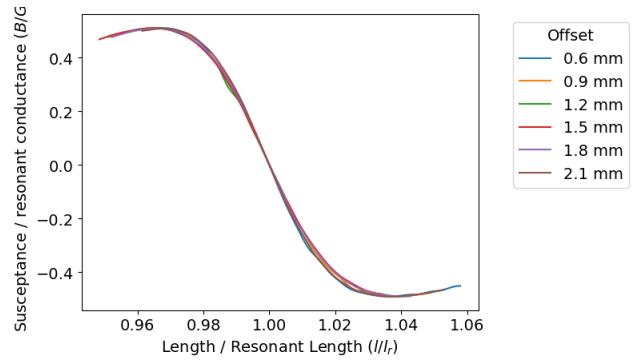


Figure 5.11: Normalized susceptance $\hat{B}(\hat{l})$ showing universal behavior across different offsets

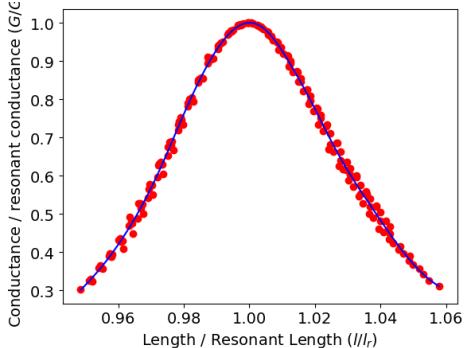


Figure 5.12: Spline interpolation of normalized conductance and susceptance for universal slot behavior modeling

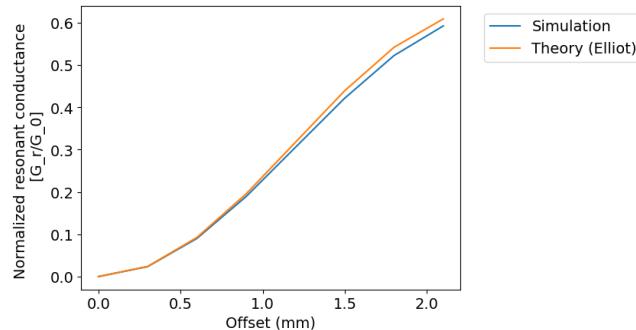


Figure 5.13: Comparison between theoretical Elliott's equation and simulated resonant conductance values

5.2.2 Single Slot Radiation Pattern

The radiation pattern of a single slot at its resonant length provides insight into the fundamental radiation characteristics. Figure 5.14 shows the elevation pattern for a slot with offset $x = 1.5$ mm and length $l = 2.5$ mm (corresponding to the resonant length for this offset).

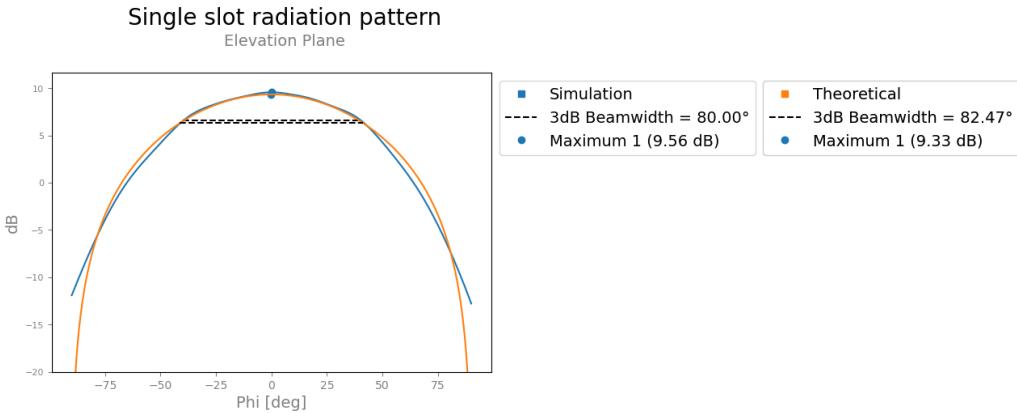


Figure 5.14: Single slot elevation radiation pattern for offset = 1.5 mm and length = 2.5 mm (resonant length)

5.2.3 Slot width variation

In figures xx and xx conductance and susceptance are respectively plotted for different values of slot width (0.1mm , 0.6mm and 1mm). Offset is left fixed at 1mm .

It is observed that as slot width increases, resonant length decreases. In the same way, as the width becomes bigger, both susceptance and conductance become flatter, meaning that the peaks are less steep and the dependence on the slot length is lower. This implies that higher widths are more insensitive to fabrication length variations.

5.2.4 Waveguide wall width variation

5.2.5 Slot characterisation for non ideal case

In this section, the previously obtained methodology will be repeated for the non-ideal case ($h = 0.5\text{mm}$, $w = 0.5\text{mm}$).

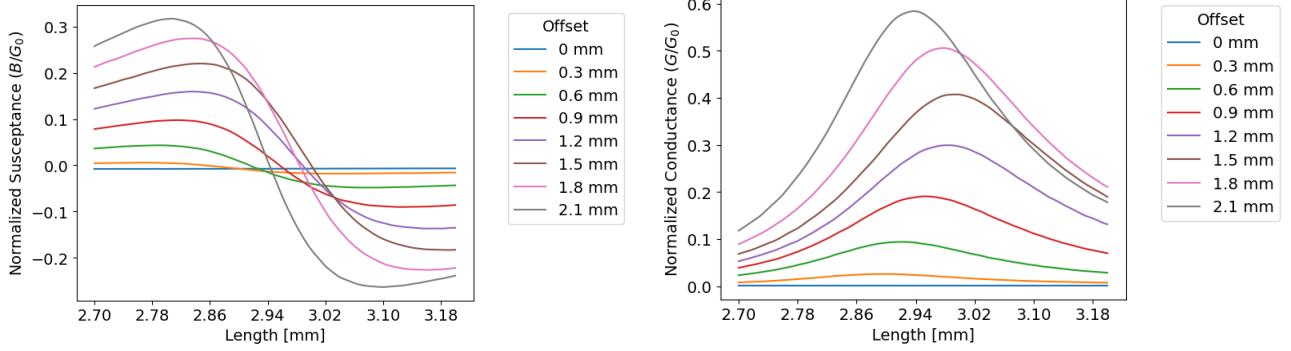


Figure 5.15: Simulated conductance (G) vs slot length for different offsets (non-ideal case: $h=0.5\text{mm}$, $w=0.5\text{mm}$)

Figure 5.16: Simulated susceptance (B) vs slot length for different offsets (non-ideal case: $h=0.5\text{mm}$, $w=0.5\text{mm}$)

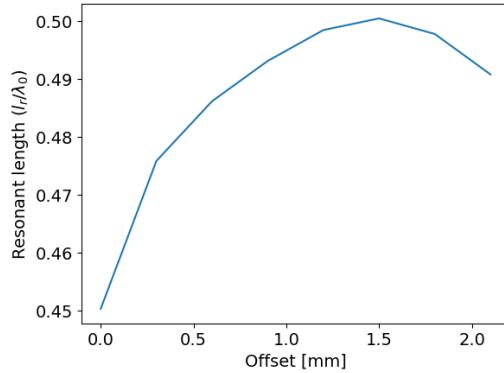


Figure 5.17: Resonant length variation with offset for non-ideal slot geometry

We can then once again build normalised susceptance and conductance functions, shown in figure 5.18.

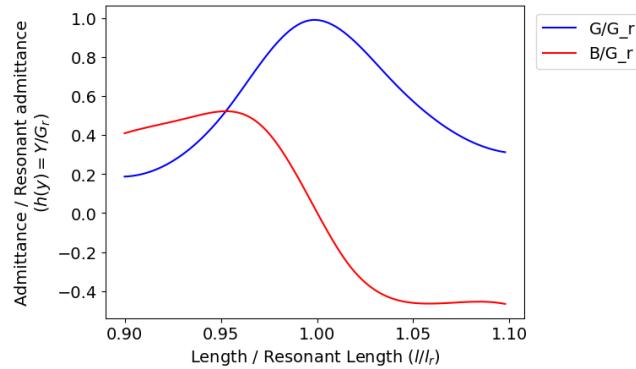


Figure 5.18: Simulated Susceptance (B)

5.3 Antenna design

We will look forward to design an antenna with 10 slots, so as to follow a Chebyshev distribution given by figure xx.

5.3.1 Slot voltages

The slot voltages V_s are chosen so as to follow a Chebyshev distribution. the SLL (side lobel level) is chosen so as to achieve the field of view of around 20 at $-3dB$ as per the specifications. This happens because higher SLL are associated to a narrower FoV. The obtained array factor is given in image xx.

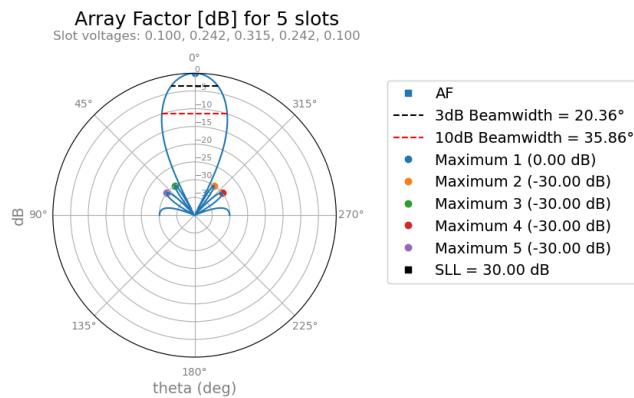


Figure 5.19: Theoretical array factor for Chebyshev distribution with optimized side lobe level for 20° field of view

5.3.2 Ideal case

Without coupling considerations

Slot	1	2	3	4	5
Offset (mm)	0.3977	-1.0521	1.5058	-1.0521	0.3977
Length (mm)	2.7356	2.8010	2.8211	2.8010	2.7356

Table 5.1: Offsets and Lengths in mm for Each Slot (Without Coupling Considerations)

With coupling considerations

Slot	1	2	3	4	5
Offset (mm)	0.3977	-0.9521	1.4058	-0.9521	0.3977
Length (mm)	2.7671	2.8119	2.8364	2.8119	2.7671

Table 5.2: Offsets and Lengths in mm for Each Slot with Coupling Considerations

Elevation radiation pattern

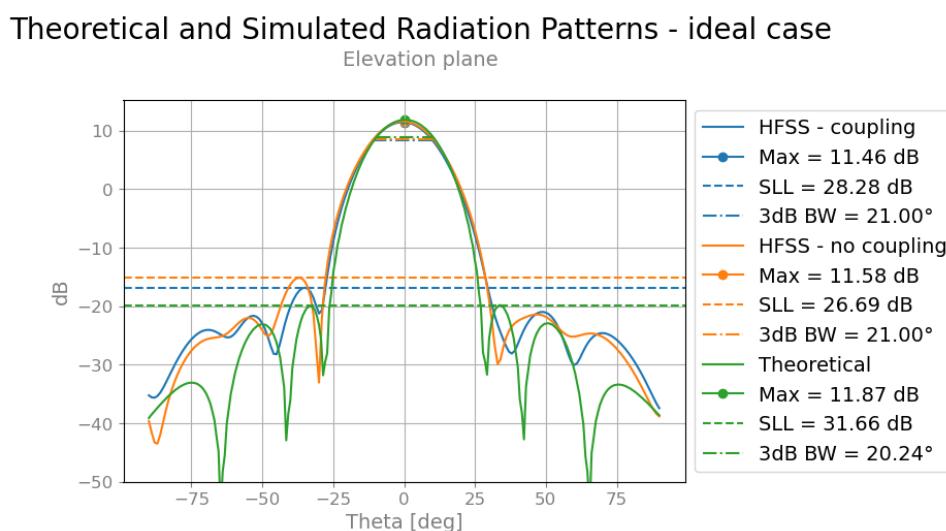


Figure 5.20: Elevation radiation pattern results for ideal 5-slot SWA with Chebyshev distribution

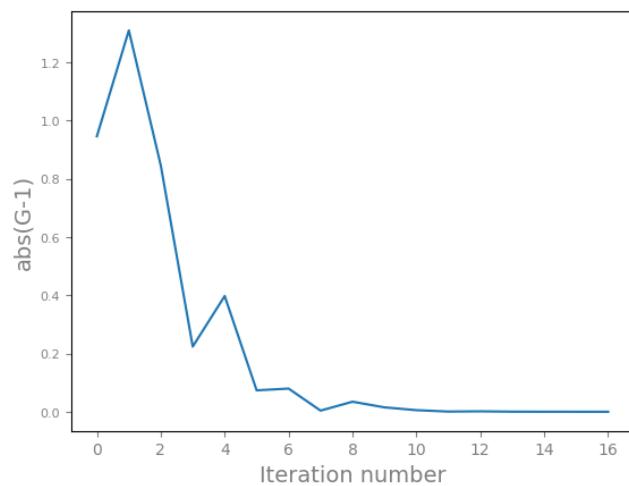


Figure 5.22: Enter Caption

Azimuth radiation pattern

Theoretical and Simulated Radiation Patterns - ideal case
Azimuth plane

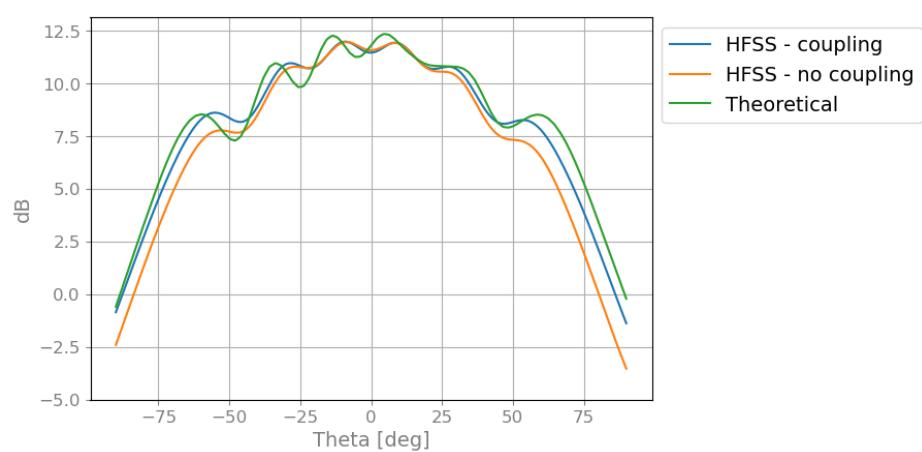


Figure 5.21: Azimuth radiation pattern results for ideal 5-slot SWA with Chebyshev distribution

S11 results

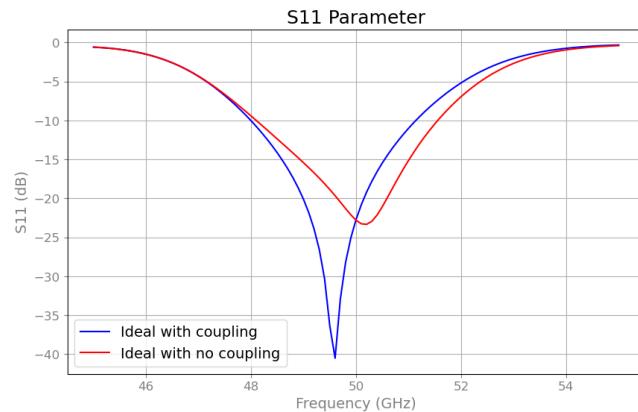


Figure 5.23: Enter Caption

5.3.3 Non ideal case

Without coupling considerations

Slot	1	2	3	4	5
Offset (mm)	0.3749	-1.0520	1.5797	-1.0520	0.3749
Length (mm)	2.3736	2.4744	2.4990	2.4744	2.3736

Table 5.3: Offsets and Lengths in mm for Each Slot (Original Values)

With coupling considerations

Slot	1	2	3	4	5
Offset (mm)	0.3749	-0.9520	1.4797	-0.9520	0.3749
Length (mm)	2.4152	2.4838	2.5165	2.4838	2.4152

Table 5.4: Offsets and Lengths in mm for Each Slot with Coupling Considerations

Elevation radiation pattern

Theoretical and Simulated Radiation Patterns - non-ideal case
Elevation plane

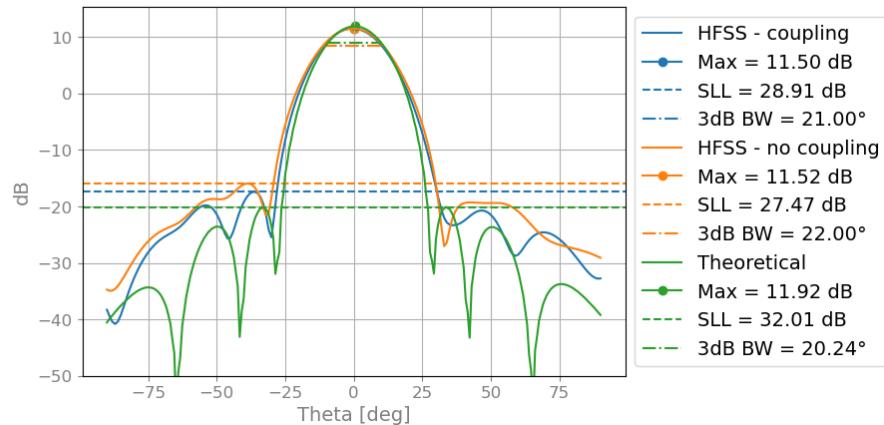


Figure 5.24: Enter Caption

Azimuth radiation pattern

Theoretical and Simulated Radiation Patterns - non-ideal case
Azimuth plane

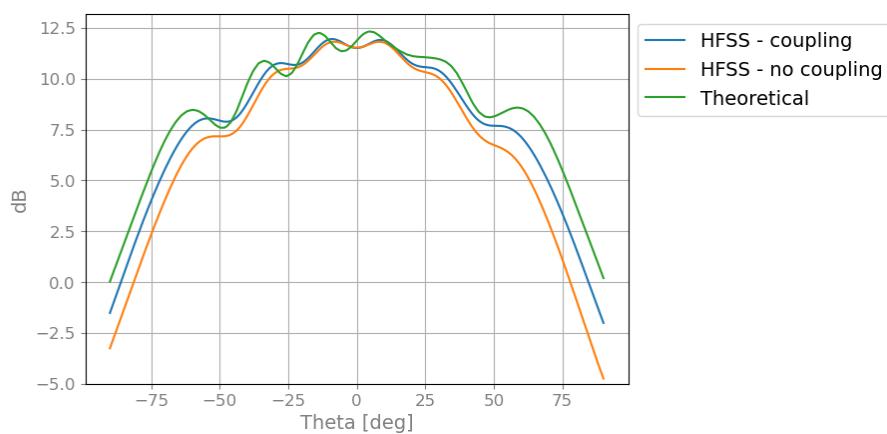


Figure 5.25: Enter Caption

S11 results

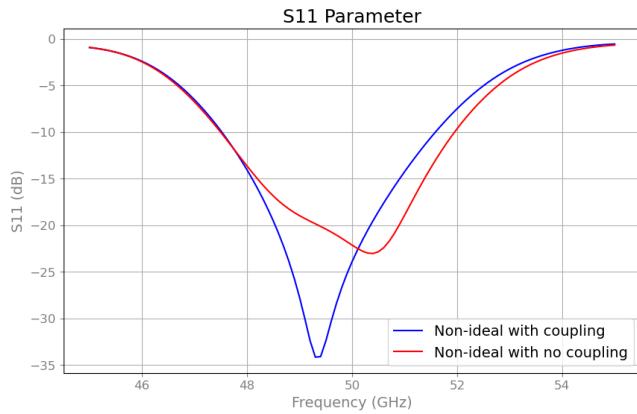


Figure 5.26: Enter Caption

5.4 Cross and co-Polarization

Cross-polarization is a measure of how much power is being radiated in non-horizontal E field directions. The typical 3D cross-polarization HFSS diagram is shown in image 5.27. It is a good measure of how much power is being lost to an unwanted polarization when a linear polarization is searched. Copolarisation refers then to the radiation pattern of the other E field direction.

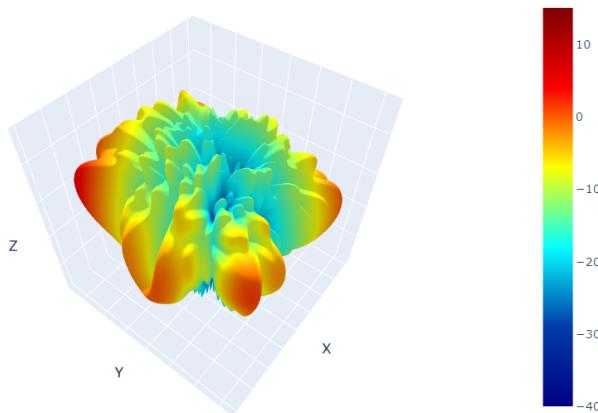


Figure 5.27: Enter Caption

Logically, the radiation patterns should verify the following relationship:

$$P_{\text{copol}} + P_{\text{cross}} = P_{\text{total}} = 4\pi$$

5.5 Optimization on the 3D Radiation Pattern

While the results obtained so far confirm the correct behavior of the Elliot algorithm, new challenges emerge when considering the full 3D radiation pattern.

Figure 5.28 presents both simulation and real-world measurements for a specific SWA design, demonstrating good agreement and confirming the reliability of the simulation results. However, in both cases, unwanted side-lobes appear in the diagonal directions. This suggests that optimizing only along the vertical and horizontal cuts may be insufficient, and a more comprehensive 3D optimization strategy should be considered.

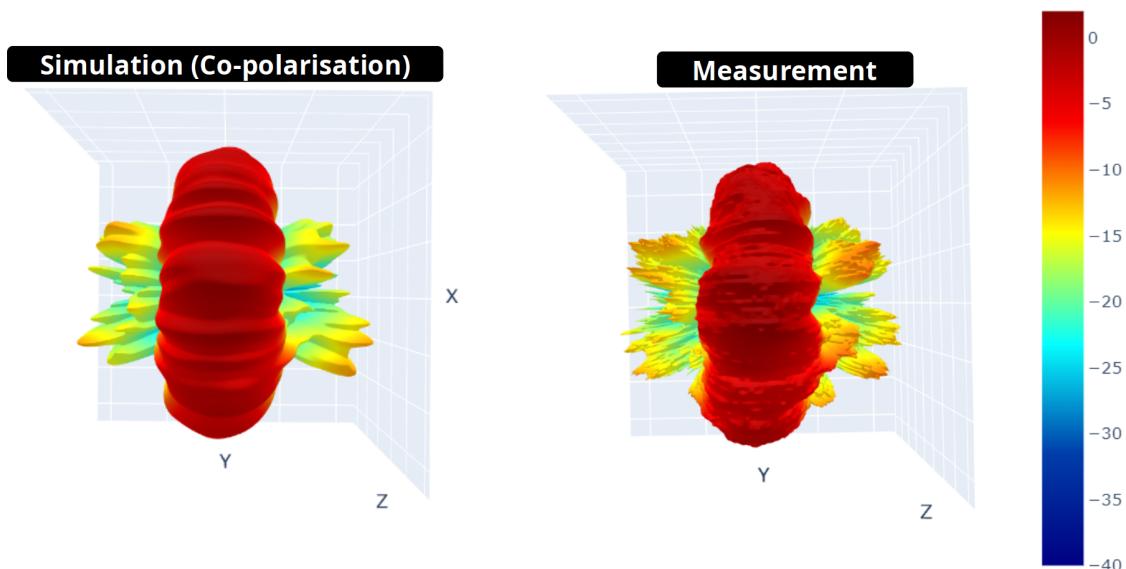


Figure 5.28: Enter Caption

In all cases, the algorithm illustrated in Figure 5.29 is employed to optimize the slot voltages V_n^s , offsets x , and lengths l . The optimization procedure follows an alternating scheme: in one step, the slot voltages are optimized; in the next, the offsets and lengths are obtained. This process iterates until convergence, with respect to x . For each optimization step, the classical L-BFGS-B algorithm from SciPy in Python is utilized.

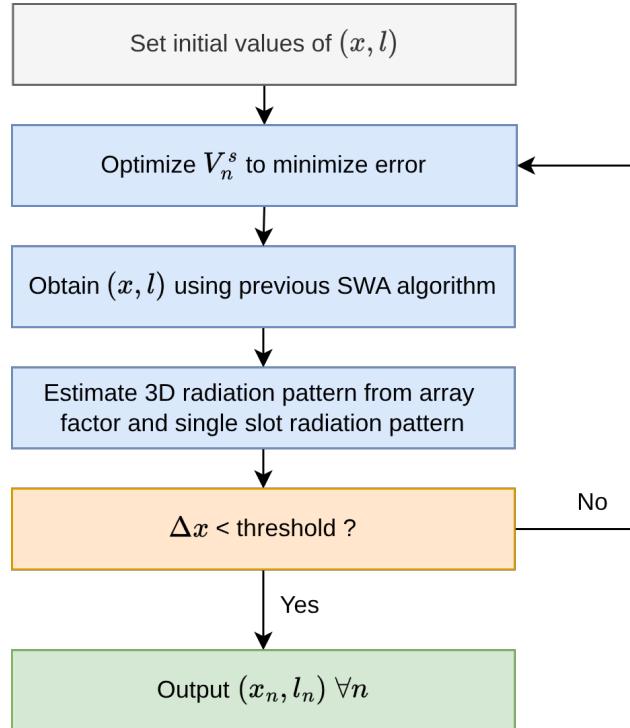


Figure 5.29: Enter Caption

The question now is what error metric to consider, for which two possibilities are explained in the following subsection.

5.5.1 Optimization based on out of FoV power

As a first approach, one could try to compute how much power is outside the FoV, and try to minimize this metric. The out of FoV radiation pattern is defined as

$$P_{FoV}(\theta, \phi) = P(\theta, \phi) \text{ if } \theta_a < \theta_{-3dB}$$

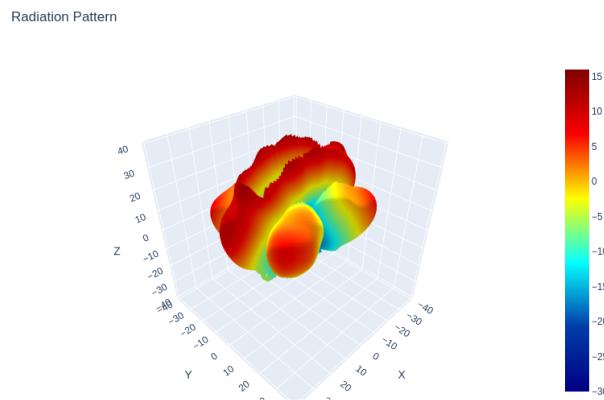


Figure 5.30: out of FoV radiation pattern

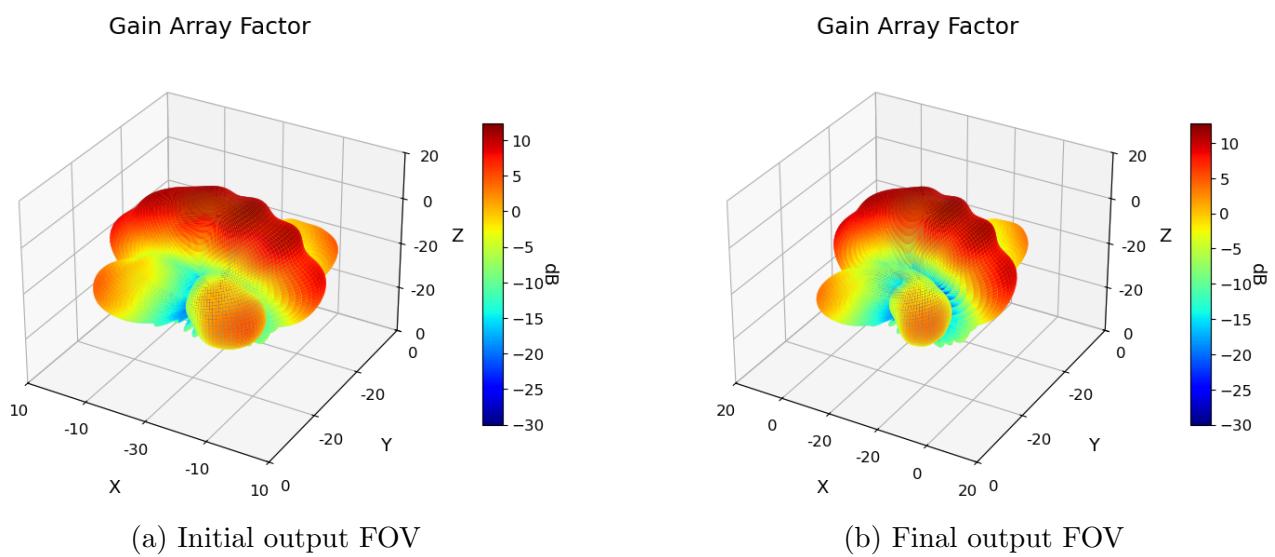


Figure 5.31: Comparison of initial vs. final field of view outputs.

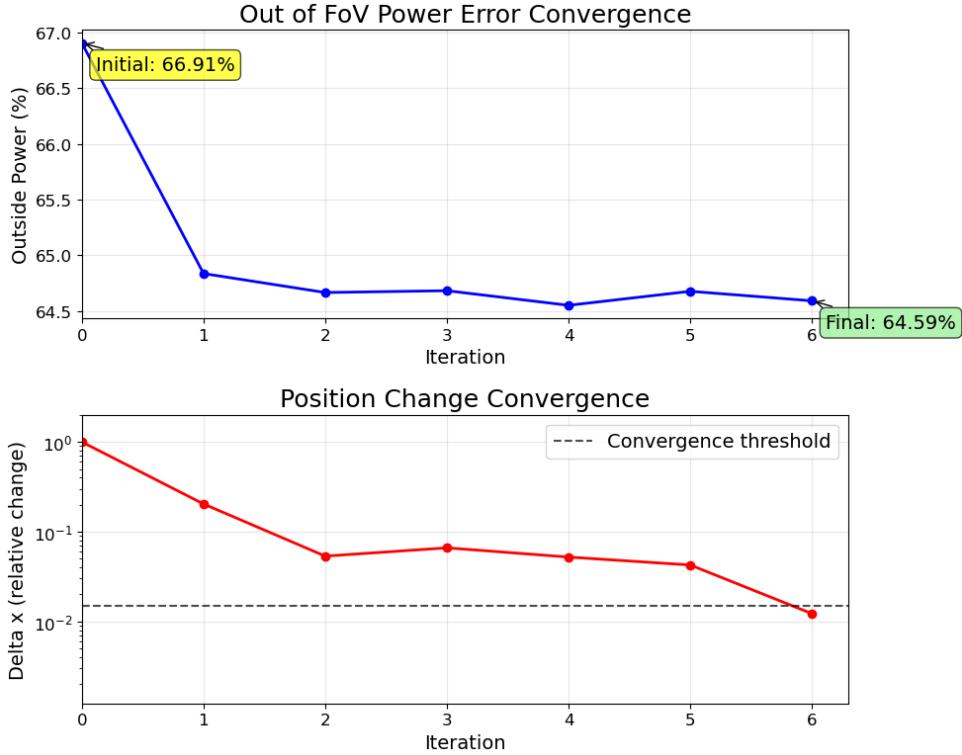


Figure 5.32: Enter Caption

5.5.2 Optimization based on a target radiation pattern

As a solution to the forementioned problem, an alternative optimization methodology based upon a target "ideal" radiation pattern is proposed. Firstly, an imperative question must be answered: which is the target radiation pattern to be obtained? Usually, one could consider that there are some specifications to be respected, such as the $-3dB$ and $-10dB$ power levels, for both perpendicular cuts of the radiation pattern (that is, azimuth and elevation planes). These points define 8 points of the target radiation pattern, and by, adding the restriction that the 3D radiation pattern must be smooth at this points, an adequate function can be proposed to verify the specifications.

For instance, the following function will be used, where the target radiation pattern is modeled as an exponential decay function:

$$P_{target}(\theta_a, \varphi_a) = \exp \left(- \left| \frac{\varphi_2}{\pi} \cdot 180 \right|^{\alpha_\varphi} / \sigma_\varphi - \left| \frac{\theta_2}{\pi} \cdot 180 \right|^{\alpha_\theta} / \sigma_\theta \right) \quad (5.11)$$

Here, the transformed angles θ_a, φ_a are a function of the spherical angles θ_s, φ_s given in Section ???. However, to ensure that this radiation pattern can be easily compared to the ones produced by the algorithms involved, spherical coordinates (θ_s, φ_s) will be used to define the

points, and the conversion formula between a spherical coordinate system and the alternative coordinate system, presented in Section ??, will be used.

The following definitions are introduced:

- $\alpha_\varphi, \alpha_\theta$ are the order parameters controlling the shape.
- $\sigma_\varphi, \sigma_\theta$ are the scaling parameters affecting the width.
- The angles θ_2, φ_2 are converted to degrees for calculation.

θ and φ are the angles associated to a spherical coordinate system. To find the parameters $\alpha_\varphi, \alpha_\theta, \sigma_\varphi$ and σ_θ an optimization is performed using python libraries, in this case the algorithm of *differential evolution* of `scipy.optimize`. The objective function to minimize is the sum of the squared differences between the current -3 dB and -10 dB markers and the specified ones. Once optimization is performed, the following values have been obtained for the parameters:

$$\sigma_\theta = 150232.97, \quad \alpha_\theta = 3.28 \quad (5.12)$$

$$\sigma_\varphi = 6241.75, \quad \alpha_\varphi = 3.44 \quad (5.13)$$

Figure 5.35 shows the resulting target radiation pattern and its verification against the specified beam characteristics.

Target Radiation Pattern

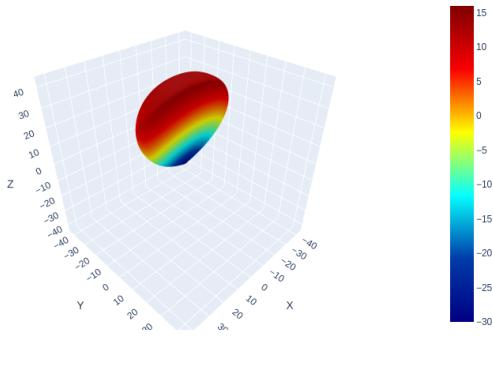


Figure 5.33: Target 3D radiation pattern optimized for radar specifications

2D Radiation Diagrams
Phi = 0 and Phi = 90

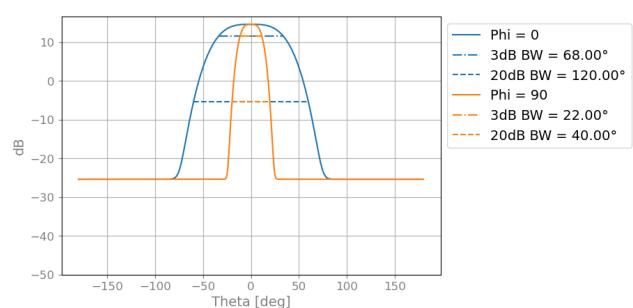


Figure 5.34: 2D slices showing -3 dB and -10 dB specifications verification

Figure 5.35: Target 3D radiation pattern (left) and -3 dB and -10 dB specifications verification (right)

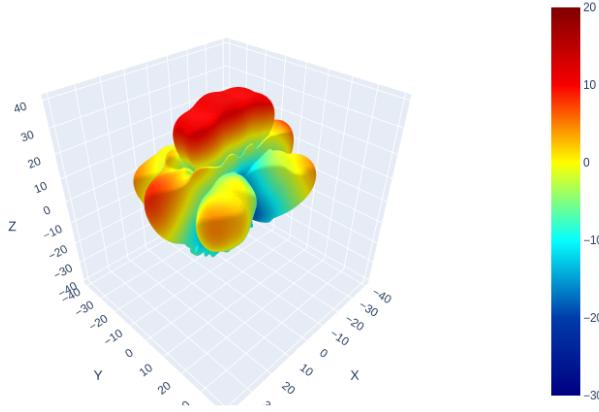


Figure 5.36: Absolute difference between target radiation pattern and estimated radiation pattern

Once the target radiation pattern is defined, it is possible to measure the difference, point by point, between this target and the obtained radiation pattern. In this way, the absolute value of the error can be obtained, as depicted in Figure 5.36.

To extract a single-valued metric from this 3D error plot, it is proposed to work with the power of this absolute error, which shall be computed as

$$\text{Error power} = \int_{\varphi} \int_{\theta} |P(\theta, \varphi) - P_{target}(\theta, \varphi)| \sin(\theta) d\theta d\varphi \quad (5.14)$$

Having established a metric for how well our design fits to the 3D target radiation pattern specification, it is of interest to modify the previously used algorithm to incorporate this variable. In Figure 5.37, the optimization process shows convergence toward the target specifications. The idea consists of optimizing in an alternating manner slot amplitude values (V_n), and then positions and offsets on the next step. While convergence is not proved in this text, results have been able to reach at least a local minimum.

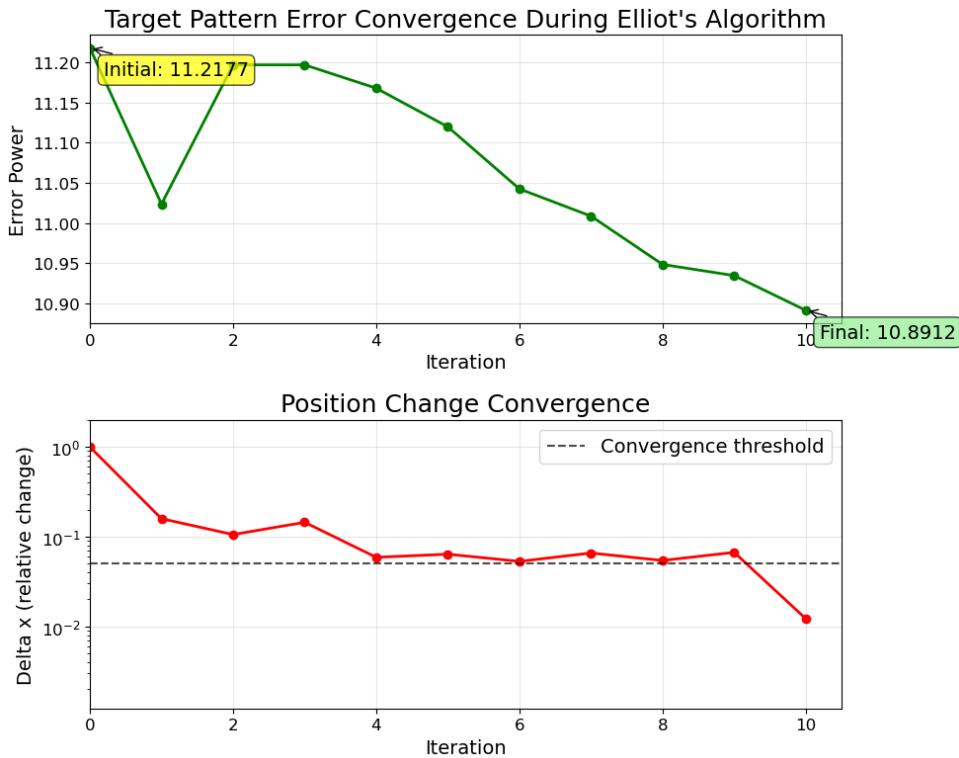


Figure 5.37: Convergence behavior of the target pattern optimization algorithm showing improvement over iterations

5.5.3 Azimuth Plane beam shaping

While significant improvements can be obtained through intensive optimisation algorithms, it is of interest to know that there are alternatives to reduce the FoV on the Azimuth plane, which is linked to the x (offset) dimension. For instance, in Figure ??, an innovative approach using several waveguides, which are fed by a lower waveguide with longitudinal slots, is given.

Chapter 6

Non-resonant Slot SWA Optimization

In this chapter, we analyze non-resonant slots in a standing-wave slotted-waveguide array (SWA). By introducing extra geometric degrees of freedom in the antenna design, we enable more refined and sophisticated optimizations. First, we present the theoretical framework for solving a system of non-resonant slots; next, we detail a Genetic Algorithm (GA) procedure that exploits this model; and finally, we discuss a design example alongside its validation via HFSS simulation.

6.1 Theoretical Model

Let each offset have variable position p_n , offset x_n and length l_n , and fixed width $w = 0.5\text{mm}$. Assuming each slot corresponds to a shunt admittance, the transmission line model depicted in Figure ?? will be used.

6.1.1 ABCD Parameters

A two-port network representation of a transmission-line segment of electrical length ℓ and propagation constant β is given by its ABCD matrix:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}_{\text{TL}} = \begin{bmatrix} \cos(\beta\ell) & jZ_0 \sin(\beta\ell) \\ jY_0 \sin(\beta\ell) & \cos(\beta\ell) \end{bmatrix},$$

where Z_0 and $Y_0 = 1/Z_0$ are the characteristic impedance and admittance respectively. Since we assume impedances and admittances to be normalized, we will actually have $Z_0 = 1$ and $Y_0 = 1$. A shunt admittance Y (e.g. modeling the slot's equivalent circuit) is represented by

$$\mathbf{M}_{\text{shunt}}(Y_n) = \begin{bmatrix} 1 & 0 \\ Y_n & 1 \end{bmatrix} \quad (6.1)$$

Since these matrices are cascaded along the guide, the overall ABCD parameters are obtained by ordered multiplication of the per-segment matrices.

6.1.2 Wave Propagation in a Transmission Line with Shunt Elements

Consider a waveguide whose aperture slots introduce localized shunt admittances Y_n . Let the positions of these slots be p_n along the guide, and denote the total electrical length to the load termination by $L = 3.25\lambda_g$. The cumulative ABCD matrix from the input port to the load is

$$\mathbf{M}_{\text{tot}} = \prod_{n=0}^N \left[\mathbf{M}_{\text{TL}}(p_{n+1} - p_n) \mathbf{M}_{\text{shunt}}(Y_n) \right],$$

with $p_0 = 0$ and $p_{N+1} = L$. From $\mathbf{M}_{\text{tot}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ one recovers the global input impedance

$$Z_{\text{in}} = \frac{AZ_L + B}{CZ_L + D} \quad \text{with} \quad Y_{\text{in}} = \frac{1}{Z_{\text{in}}},$$

where for an ideal short-circuited end $Z_L = 0$, reducing to $Z_{\text{in}} = B/D \Rightarrow Y_{\text{in}} = D/B$.

6.1.3 Waveguide Voltage Calculation

To extract the waveguide voltage at each slot position, we employ the inverse ABCD transformation. Given the input voltage and current at the beginning of each transmission line segment, the voltage and current at the output (slot location) are found by applying:

$$\begin{bmatrix} V_{\text{out}} \\ I_{\text{out}} \end{bmatrix} = \mathbf{M}_{\text{TL}}^{-1} \begin{bmatrix} V_{\text{in}} \\ I_{\text{in}} \end{bmatrix} \quad (6.2)$$

This process is repeated sequentially for each slot position, where the output voltage and current from one segment become the input for the next segment after accounting for the current drawn by the shunt admittance. The waveguide voltage V_n at slot n corresponds to the voltage calculated just before the shunt admittance element.

6.1.4 Slot admittance characterization

While we previously normalized to model the base (uncoupled) admittances, we will now employ the raw interpolated simulation data. As shown in Figure 6.1, the resonant region—where conductance peaks and susceptance crosses zero—is highly nonlinear. Henceforth, during optimization we will directly extract the admittance $Y(x, l)$ from this interpolation for each slot coordinate pair (x_n, l_n) .

In that resonant zone, susceptance jumps abruptly from negative to positive, which exacerbates the antenna's statistical (Monte Carlo) sensitivity to manufacturing tolerances. Avoiding this unstable region is therefore a key motivation for using non-resonant slots.

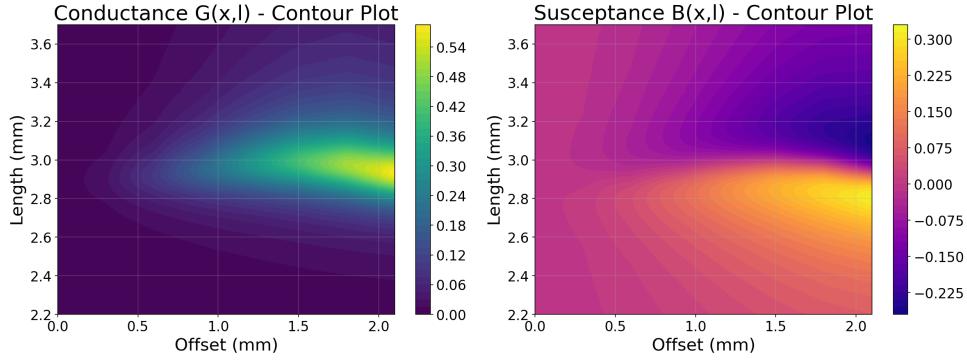


Figure 6.1: Slot admittance values as a function of offset and length

6.1.5 Design Equations

As per Elliot theory, the following design equations must hold:

$$\frac{Y_n^a}{G_0} = \frac{2 f^2(x_n, l_n)}{\frac{2 f^2(x_n, l_n)}{Y(x_n, l_n)/G_0} + j \frac{\beta_{10}}{k} k_0 b \left(\frac{a}{\lambda}\right)^3 \sum_{m=1}^n \frac{V_m^s}{V_n^s} g_{mn}} \quad (6.3)$$

$$\frac{Y_n^a}{G_0} = K_1 f(x_n, l_n) \frac{V_n^s}{V_n} \quad (6.4)$$

Finally, as shown in Section 6.1.3, the actual waveguide voltage at each slot is determined from its equivalent admittance and position:

$$V_n = h_n(\mathbf{Y}, \mathbf{p}), \quad (6.5)$$

where $h_n()$ is the result of cascading ABCD matrices as previously shown.

6.2 Antenna solver Algorithm

Overall, this system defines a system of $3N$ equations for $3N$ variables: \mathbf{V} , \mathbf{V}^s and \mathbf{Y}^a . However, the system will be simplified by only optimising over V_n^s

6.2.1 Self-Consistent Solution Algorithm

The antenna solver algorithm implements a self-consistent numerical solution to Elliott's design equations. For a given slot geometry $(\mathbf{x}, \mathbf{l}, \mathbf{p})$, the system seeks slot voltages \mathbf{V}^s that simultaneously satisfy all coupling relationships.

Problem Formulation

The solver transforms the complex system of equations into a root-finding problem by defining a residual function $\mathbf{F}(\mathbf{V}^s) = \mathbf{0}$. For each slot n , the residual is:

$$F_n(\mathbf{V}^s) = V_n^{s,\text{guess}} - V_n^{s,\text{calculated}} \quad (6.6)$$

where $V_n^{s,\text{calculated}}$ is computed from the design equations:

$$V_n^{s,\text{calculated}} = \frac{Y_n^a \cdot V_n}{K_1 \cdot f(x_n, l_n)} \quad (6.7)$$

Iterative Solution Process

The algorithm follows this iterative sequence for each guess of \mathbf{V}^s :

1. **Admittance Calculation:** For each slot n , compute the active admittance using direct interpolation from simulation data:

$$Y_n^a = \frac{2f^2(x_n, l_n)}{\frac{2f^2(x_n, l_n)}{Y(x_n, l_n)} + \text{MC}_n(\mathbf{V}^s)} \quad (6.8)$$

2. **Waveguide Voltage Computation:** Calculate waveguide voltages \mathbf{V} using cascaded ABCD matrices based on the computed admittances \mathbf{Y}^a .
3. **New Slot Voltage Calculation:** Compute new slot voltages using Equation 6.4.
4. **Residual Evaluation:** Calculate the residual vector $\mathbf{F}(\mathbf{V}^s)$.

Numerical Solver Implementation

The system employs two alternative numerical methods:

- **MINPACK Hybrid Method:** Uses the `scipy.optimize.root` function with `method='hybr'` for robust convergence.
- **Trust Region Method:** Uses `scipy.optimize.least_squares` with `method='trf'` and soft-L1 loss for improved exploration.

Initial Guess Strategy

The algorithm generates an initial guess by solving the uncoupled system (setting mutual coupling terms to zero):

$$V_n^{s,\text{initial}} = \frac{Y_n^{\text{uncoupled}} \cdot V_n^{\text{uncoupled}}}{K_1 \cdot f(x_n, l_n)} \quad (6.9)$$

Zeroed Slot Recovery Mechanism

An issue may arise with the proposed optimization method: the algorithm can erroneously converge to a solution in which a slot has zero active admittance (Y_n^a) and zero slot voltage (V_n^s). Such local minima effectively remove the slot from the system—they neither affect the voltage equation (6.5), since their equivalent shunt admittance is zero, nor contribute to mutual coupling in the external coupling equation (6.8), as their slot voltage vanishes. In practice, this degeneracy causes the algorithm to behave as though it is designing an antenna with fewer slots.

To improve consistency and prevent this pathological behavior, the algorithm incorporates a two-part recovery strategy with distinct mechanisms for individual slot recovery and global restart:

Individual Slot Recovery

1. Identify converged solutions where any slot falls below the activity threshold ($|V_n^s| < V_{\text{threshold}}^s$ and $|Y_n^a| < Y_{\text{threshold}}$).
2. For each inactive slot, assign a new random complex excitation value:

$$V_{\text{zeroed}}^s = |V_{\text{rand}}| \cdot e^{j\phi_{\text{rand}}} \quad (6.10)$$

where uniform random variables $|V_{\text{rand}}| \in [V_{\min}, V_{\max}]$ and $\phi_{\text{rand}} \in [0, 2\pi]$ are used.

3. Restart the optimization using this modified initial guess while preserving active slot values.
4. Repeat this individual slot recovery process up to N_{recovery} times per solution attempt.

Global Multi-Start Recovery

If individual slot recovery fails to produce a solution with all active slots:

1. Generate completely new random initial guesses for all slot voltages with amplitude $\pm A_{\text{global}}$.
2. Restart the entire optimization process from scratch.

3. Repeat this global restart process up to N_{attempts} times with different random initializations.
 4. Return the best solution found across all attempts, ranked by convergence status and number of active slots.

where the algorithm parameters are: $V_{\text{threshold}}^s = Y_{\text{threshold}} = 10^{-6}$, $V_{\min} = 0.05$, $V_{\max} = 0.5$, $N_{\text{recovery}} = 10$, $A_{\text{global}} = 2.0$, and $N_{\text{attempts}} = 50$.

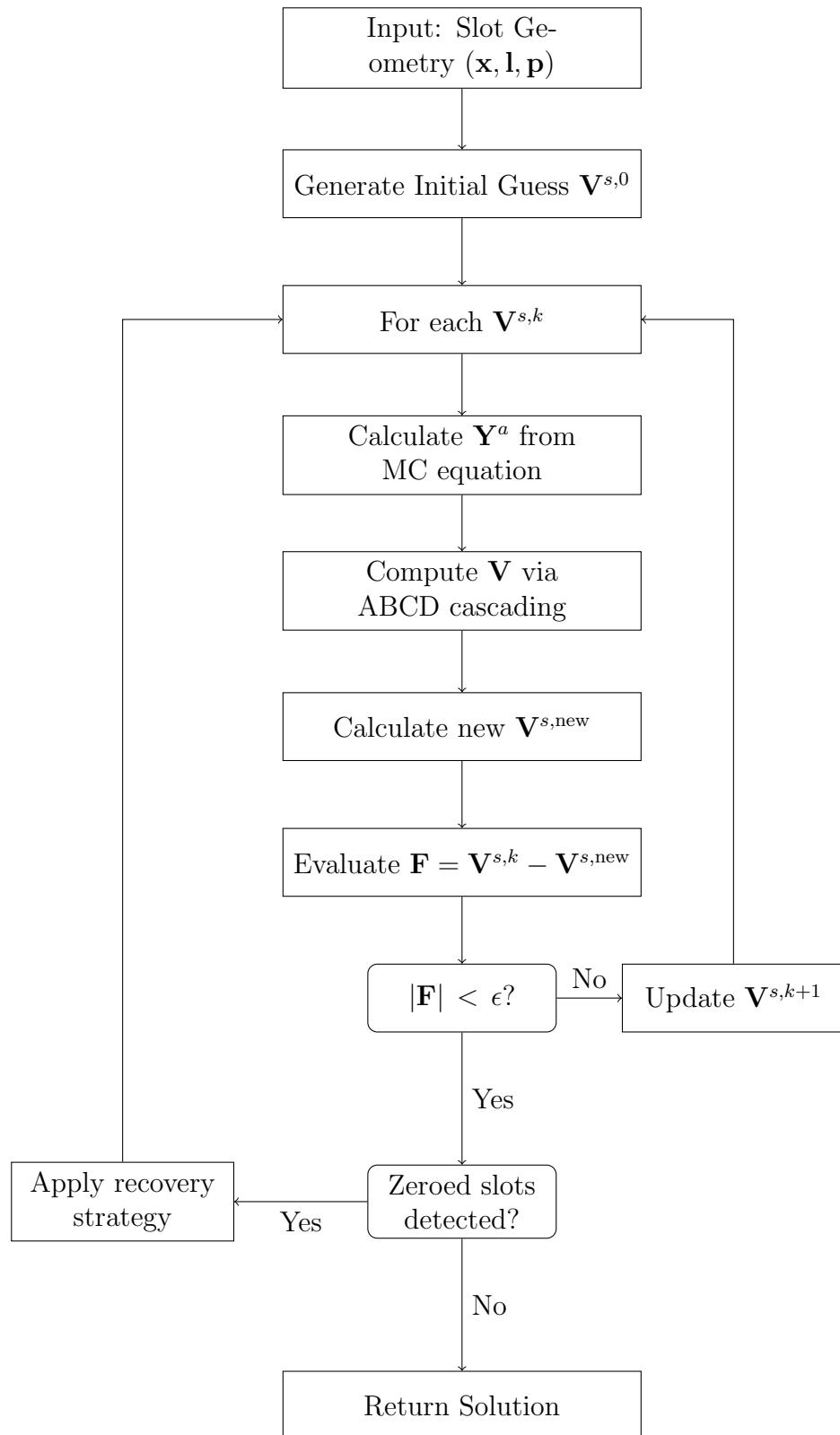


Figure 6.2: Antenna solving algorithm flowchart

6.3 Genetic Algorithm for Slot Array Optimization

The genetic algorithm (GA) framework provides a robust metaheuristic approach for optimizing the slot array geometry while respecting complex geometric constraints and leveraging the antenna solver's capabilities. This section details the GA implementation, including constraint handling, algorithmic components, and parallelization strategies employed to achieve effective global optimization.

6.3.1 Geometry Constraint Processing

The optimization process operates within a constrained design space defined by physical and electromagnetic requirements. The constraint system encompasses three primary categories:

Dimensional Bounds

Each slot parameter is bounded by physically realizable limits:

$$x_{\min} \leq x_n \leq x_{\max} \quad (\text{slot positions}) \quad (6.11)$$

$$o_{\min} \leq o_n \leq o_{\max} \quad (\text{offset from centerline}) \quad (6.12)$$

$$l_{\min} \leq l_n \leq l_{\max} \quad (\text{slot lengths}) \quad (6.13)$$

where bounds are expressed relative to the guided wavelength λ_g and waveguide dimensions to ensure electromagnetic and manufacturing feasibility.

Non-Overlap Constraints

Slots must maintain sufficient separation to prevent physical overlap and electromagnetic interference. For any slot pair (i, j) , the constraint is:

$$\text{distance}_{ij} \geq w_{\text{eff}} + 2p_{\text{safety}} \quad (6.14)$$

where w_{eff} represents the effective slot width including electromagnetic fringing fields, and p_{safety} is a safety margin accounting for manufacturing tolerances.

6.3.2 Genetic Algorithm Components

The GA implementation employs standard evolutionary operators adapted for the constrained slot optimization problem:

Individual Representation

Each individual encodes a complete slot array geometry as a real-valued vector:

$$\mathbf{g} = [x_1, x_2, \dots, x_N, o_1, o_2, \dots, o_N, l_1, l_2, \dots, l_N]^T \quad (6.15)$$

This representation facilitates direct application of real-valued genetic operators while maintaining clear physical interpretation.

Selection Mechanism

Tournament selection with replacement chooses parent individuals for reproduction. This mechanism maintains selective pressure toward better solutions while preserving population diversity. The tournament size controls the balance between exploration and exploitation throughout the evolutionary process.

Crossover Operation

Uniform crossover creates offspring by randomly selecting each parameter from either parent with equal probability. This approach preserves valuable parameter combinations while generating novel geometric configurations:

$$\text{child}_i = \begin{cases} \text{parent}_1[i] & \text{if } r_i < p_{crossover} \\ \text{parent}_2[i] & \text{otherwise} \end{cases} \quad (6.16)$$

where r_i represents a uniform random variable for each parameter position. Note that the crossover operation does not ensure geometry constraint satisfaction of the child samples.

Mutation Strategy

Gaussian mutation applies parameter-specific perturbations scaled to the respective parameter ranges:

$$g'_i = g_i + \sigma_i \cdot \mathcal{N}(0, 1) \quad (6.17)$$

where σ_i represents the mutation step size adapted to each parameter's sensitivity and feasible range. Note that the mutation is performed iteratively till geometry constraint satisfaction.

Elitism and Population Management

Elite preservation maintains the best individuals across generations, ensuring monotonic improvement in the best solution quality. The population replacement strategy balances elitism with diversity maintenance to prevent premature convergence.

6.3.3 Voltage Inheritance Acceleration

A key innovation in the GA implementation is the voltage inheritance mechanism, which significantly accelerates convergence by exploiting the relationship between parent and child geometries. When creating offspring through crossover and mutation:

1. Inherit voltage initial guesses based on geometric similarity between parent(s) and child
2. Provide these enhanced initial guesses to the antenna solver

This approach typically reduces solver iterations since child geometries start with voltage distributions close to their optimal values, leveraging the physical continuity of electromagnetic solutions.

6.3.4 Population Initialization Strategy

To minimize invalid geometry generation while maintaining diversity, the algorithm employs a structured initialization approach:

1. Generate one initial valid geometry through systematic constraint satisfaction
2. Create the remaining population members by applying controlled mutations to this base geometry. Constraint satisfaction is satisfied by the mutation operation.
3. Replace any invalid geometries with additional mutations of the base design

This strategy ensures that all individuals begin from a known-feasible starting point while introducing sufficient diversity through the mutation process. The approach significantly reduces the computational overhead associated with constraint violation handling during population initialization.

6.3.5 Integration with Antenna Solver

Figure 6.3 illustrates the integration between the genetic algorithm and the antenna solver. The GA provides candidate geometries to the solver, which returns electromagnetic performance metrics used for fitness evaluation.

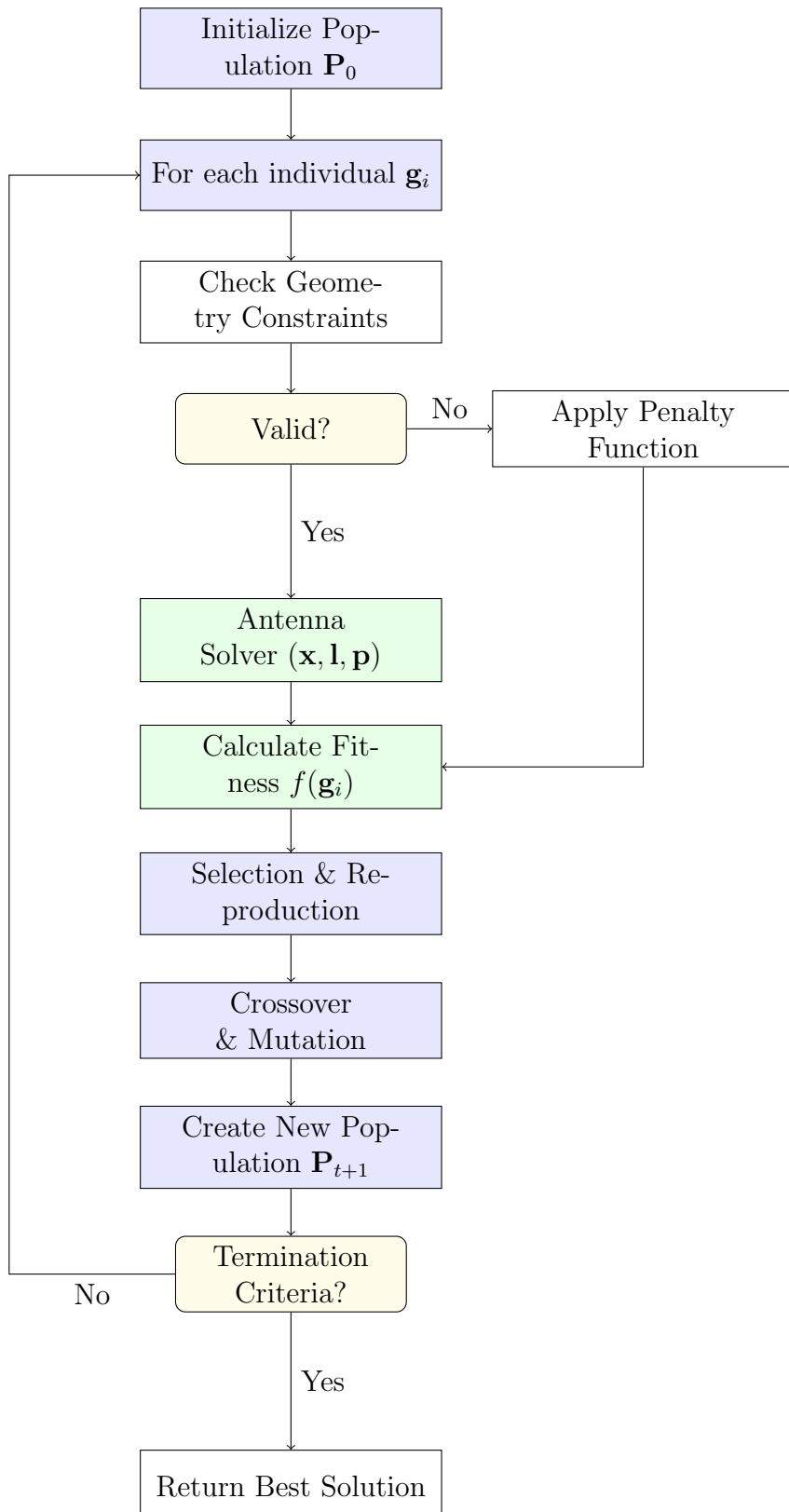


Figure 6.3: Integration flowchart showing genetic algorithm interaction with antenna solver. Blue boxes represent GA operations, green boxes represent solver operations, yellow boxes represent decision points.

The fitness function combines electromagnetic performance metrics:

$$f(\mathbf{g}) = w_{\text{adm}} \cdot L_{\text{admittance}}(\mathbf{g}) + w_{\text{rp}} \cdot L_{\text{pattern}}(\mathbf{g}) \quad (6.18)$$

where $L_{\text{admittance}}$ quantifies deviation from target input admittance and L_{pattern} measures radiation pattern error relative to the target radiation pattern.

6.3.6 Parallelization Strategy

The optimization employs a multi-level parallelization approach to maximize computational efficiency and solution quality:

Parallel Multi-Start Framework

Multiple independent GA instances execute simultaneously, each exploring different regions of the design space. This approach provides several advantages:

- **Global Exploration:** Different random initializations prevent convergence to the same local optimum
- **Robustness:** Multiple attempts increase the probability of finding high-quality solutions
- **Computational Efficiency:** Parallel execution leverages multi-core processors effectively
- **Solution Diversity:** Multiple solutions enable design trade-off analysis

Result Consolidation

Upon completion of all parallel restarts, the algorithm consolidates results by:

1. Ranking all solutions by fitness value
2. Identifying the globally optimal design
3. Analyzing solution diversity and convergence patterns
4. Generating comprehensive performance reports

This parallelization strategy effectively transforms the optimization from a single-point search into a comprehensive exploration of the design landscape, significantly improving both solution quality and robustness while maintaining computational efficiency through parallel execution.

6.4 Results

6.4.1 Hyperparameter choice

6.4.2 Algorithm evolution

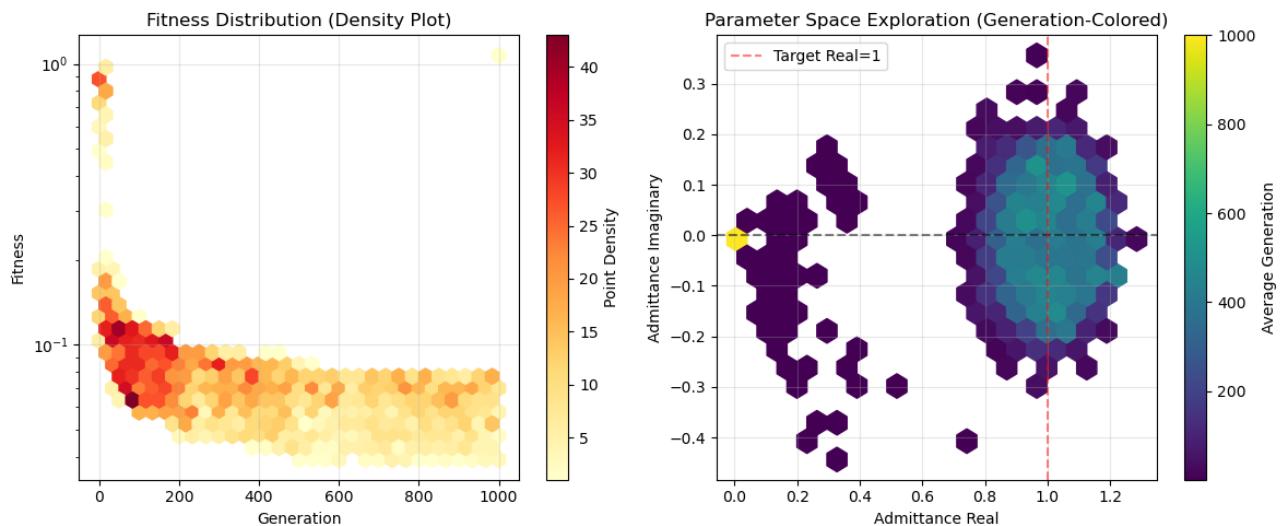


Figure 6.4: Evolution of the GA

6.4.3 Best solution

Slot	1	2	3	4	5	6	7	8
Position (mm)	3.76	4.37	7.13	9.64	9.76	12.64	15.06	15.65
Offset (mm)	-0.30	2.10	-1.85	1.38	0.08	-1.57	-0.60	1.57
Length (mm)	3.08	2.59	2.77	2.93	2.87	3.01	2.72	3.04

Table 6.1: Positions, Offsets, and Lengths in mm for Each Slot

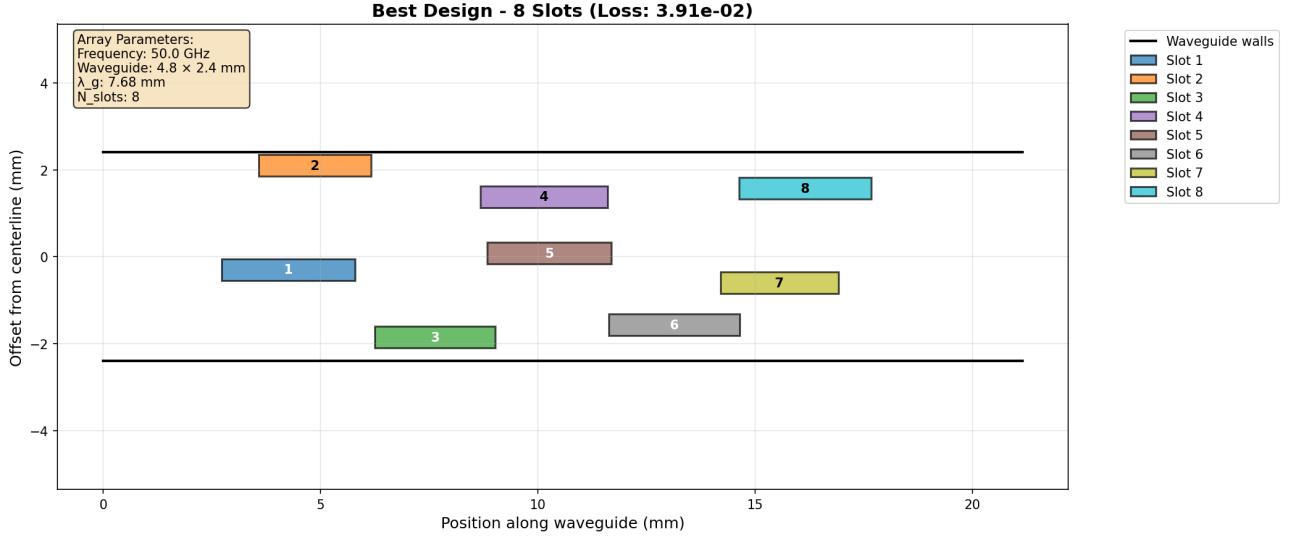


Figure 6.5: Slot disposition

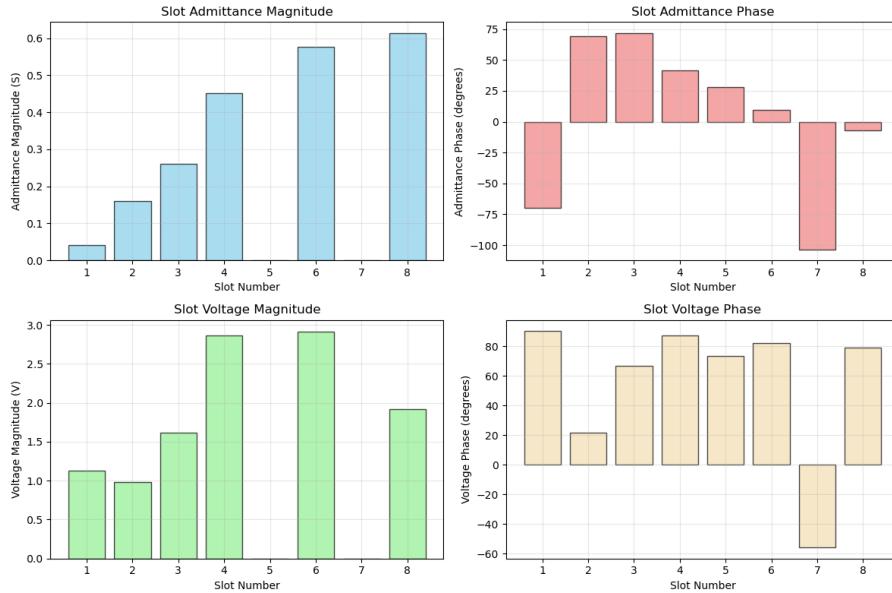


Figure 6.6: Slot amplitudes and phases

Final Error Power: 7.73 (Much lower than the error of previous designs for the same waveguide dimensions!) Final admittance: $1.0034 - 0.0207j$

Note that slots 5 and 7 are zeroed (zero admittance and slot voltage) - hence the solver couldn't find a non-zero solution and the optimization converged to a solution with 6 active slots. It is important to only simulate active slots when analysing the design in HFSS, since slots 5 and 7 do actually have a significant non-zero active admittance!

Chapter 7

Metasurface Optimisation

In this chapter, we will seek the development of an algorithm that allows to improve the SWA design from a discrete optimization method over the 2D surface that represents the antenna. This will be done by subdividing the antenna in an $N_a \times N_b$ grid, where each cell can be covered (metal, aluminium in this case) or open (air). For this matter, python scripting over HFSS will be used.

7.1 HFSS Scripting

The antenna design optimization process interfaces with ANSYS HFSS through two distinct computational approaches: a grid-based method and a vectorized approach.

7.1.1 Grid approach

The grid approach discretizes the antenna aperture into a structured mesh of individual cells, where each cell represents a small rectangular volume that can either be filled with conductor material (copper) or remain as air (slot). This discretization strategy creates a binary representation of the antenna geometry.

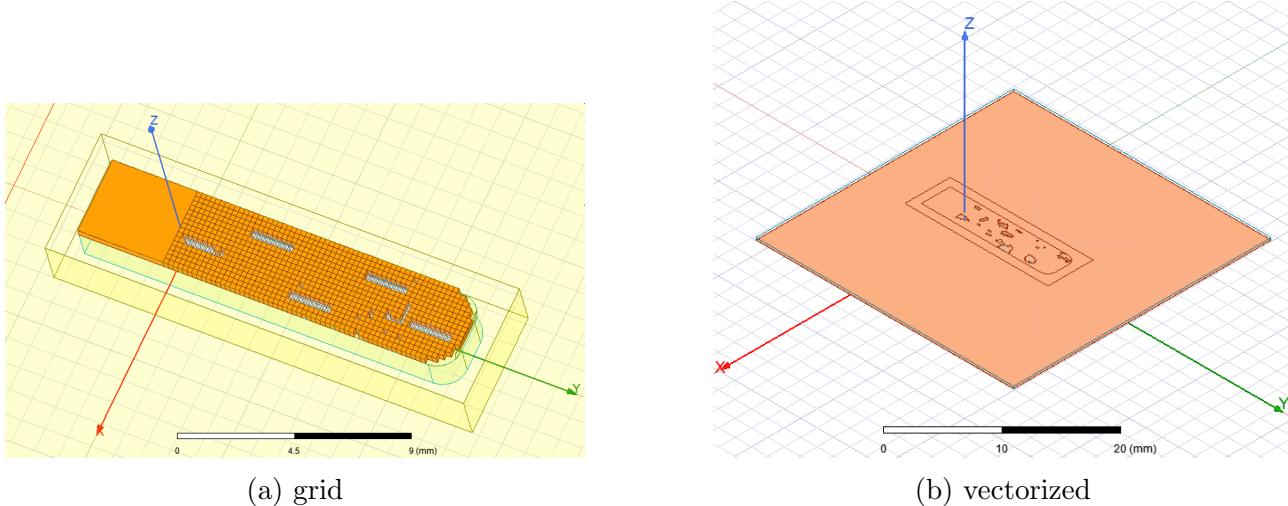


Figure 7.1: Comparison of the grid and vectorized simulations

The fundamental limitation of this approach stems from HFSS's computational complexity when handling numerous discrete geometric objects. Each cell in the grid requires individual creation, material assignment, and mesh generation within the electromagnetic simulation environment. For a typical discretization with dimensions $N_x = 48$ and $N_y = 100$, the solver must manage approximately 4,800 individual geometric entities.

This proliferation of objects creates several computational bottlenecks:

1. **Memory overhead:** Each cell requires separate data structures for geometry definition, boundary conditions, and material properties
2. **Meshing complexity:** The adaptive mesh generation algorithm must handle discontinuities at every cell boundary, significantly increasing the total number of mesh elements

Consequently, simulations become computationally prohibitive as grid resolution increases, effectively limiting the achievable design precision and constraining the optimization search space.

7.1.2 Vectorized approach

The vectorized approach addresses the computational limitations of the grid method by implementing a unified geometric representation. Instead of creating individual cells, this method generates contiguous regions of connected slots through geometric operations that merge adjacent cells with identical material properties.

The core algorithm processes the binary cell matrix to identify connected components of slot regions using flood-fill techniques. For each connected region, the method:

1. Extracts the region boundary using contour detection algorithms
2. Generates a single unified geometry object encompassing the entire slot region

3. Applies material properties to the consolidated object rather than individual cells

This vectorization dramatically reduces the number of geometric objects in the HFSS model, typically from thousands of individual cells to dozens of consolidated regions.

7.2 Random-walk optimization

The antenna design optimization employs a probabilistic random-walk algorithm that iteratively evolves the slot configuration toward improved electromagnetic performance. The algorithm maintains a probability map that guides the modification of individual cells based on their contribution to the overall antenna performance.

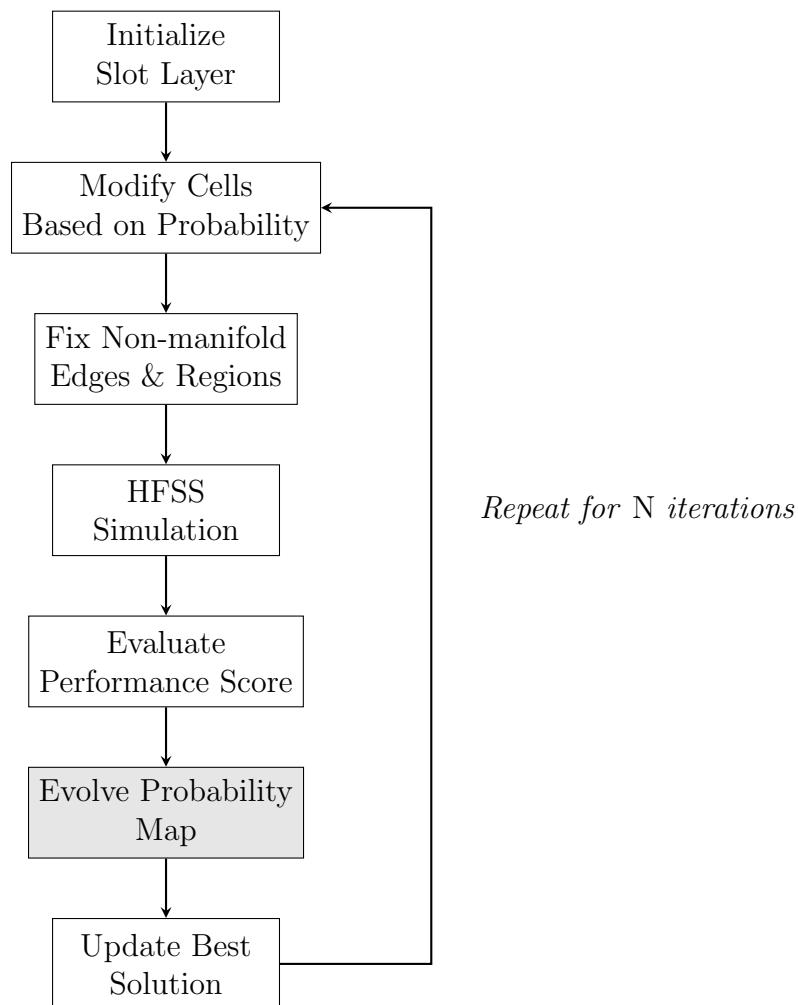


Figure 7.2: Random-walk optimization algorithm flow diagram

Note that in every iteration cell modifications are performed over the best solution found so far

(this is extremely important for the following explanation). The optimization process operates through the following key mechanisms:

Probability-driven cell modification: Each cell (i, j) in the discretized antenna aperture maintains a change probability $p_{i,j}$ that determines the likelihood of material state modification (conductor \leftrightarrow air). The probability distribution evolves based on the performance feedback from electromagnetic simulations.

Multi-stage probability adaptation: The probability update mechanism operates through three distinct stages that collectively shape the exploration strategy:

Stage 1 - Direct modification feedback: For cells that were actually modified in the current iteration (i.e., cells where the material state differs from the local best solution), the probability update follows:

$$p_{i,j}^{(k+1)} = \begin{cases} \max(\alpha_{win} \cdot p_{i,j}^{(k)}, p_{min}) & \text{if } \Delta S < 0 \text{ (improvement)} \\ \max(\alpha_{lose} \cdot p_{i,j}^{(k)}, p_{min}) & \text{if } \Delta S \geq 0 \text{ (degradation)} \end{cases} \quad (7.1)$$

where $\alpha_{win} < 1$ and $\alpha_{lose} < 1$ are learning rate parameters, and p_{min} is the minimum allowable probability.

Stage 2 - Gradient amplification: For each modified cell, the algorithm examines its four-connected neighbors and applies different probability updates based on both the performance outcome and the neighbor's material compatibility:

For neighbors who share the same material as the modified cell - after being modified - we have:

$$p_{neighbor}^{(k+1)} = \begin{cases} \max(\alpha_{win} \cdot p_{neighbor}^{(k)}, p_{min}) & \text{if } \Delta S < 0 \text{ (reinforce similarity)} \\ \min(p_{neighbor}^{(k)} \cdot (2 - \alpha_{lose}), 1) & \text{if } \Delta S \geq 0 \text{ (reduce similarity)} \end{cases} \quad (7.2)$$

For neighbors with different material from the modified cell:

$$p_{neighbor}^{(k+1)} = \begin{cases} p_{explore} & \text{if } \Delta S < 0 \text{ (encourage boundary exploration)} \\ \max(\alpha_{lose} \cdot p_{neighbor}^{(k)}, p_{min}) & \text{if } \Delta S \geq 0 \text{ (discourage changes)} \end{cases} \quad (7.3)$$

where $p_{explore}$ is a high exploration probability value.

Stage 3 - Spatial diffusion: When a new local best solution is found, the algorithm applies a diffusion mechanism that smooths probability distributions across regions of similar material. For each cell, the probability is updated as:

$$p_{i,j}^{(k+1)} = (1 - \beta) \cdot p_{i,j}^{(k)} + \beta \cdot \bar{p}_{neighbors} \quad (7.4)$$

where β is the diffusion factor and $\bar{p}_{neighbors}$ is the average probability of neighboring cells that share the same material type in the local best solution.

Border enhancement: Cells located at material boundaries (conductor-air interfaces) receive enhanced modification probabilities, recognizing their critical role in determining electromagnetic field distributions and antenna characteristics.

The algorithm incorporates several regularization mechanisms to ensure physically realizable antenna geometries, including the elimination of non-manifold edges and the enforcement of minimum connected region sizes. These constraints maintain manufacturing compatibility while preserving the electromagnetic validity of the optimized designs.

7.2.1 Results

The metasurface optimization process yields various antenna configurations. Figure 7.5 shows the final simulated antenna structure in HFSS.



Figure 7.3: Material-distribution maps: (a) initial solution, (b) optimized solution (500 iterations)

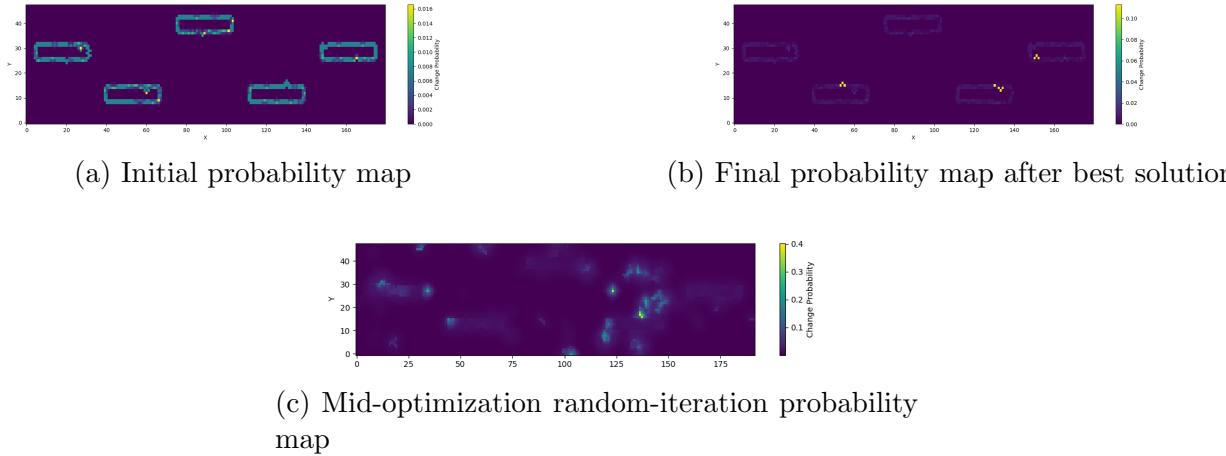


Figure 7.4: Evolution of the optimization probability map: (a) initial, (b) after finding best solution, (c) mid-optimization random iteration.

7.3 HFSS Scripting

The antenna design optimization process interfaces with ANSYS HFSS through two distinct computational approaches: a grid-based method and a vectorized approach. Each methodology

presents specific advantages and computational constraints when simulating slot waveguide antenna structures.

7.3.1 Grid approach

The grid approach discretizes the antenna aperture into a structured mesh of individual cells, where each cell represents a small rectangular volume that can either be filled with conductor material (copper) or remain as air (slot). This discretization strategy creates a binary representation of the antenna geometry.

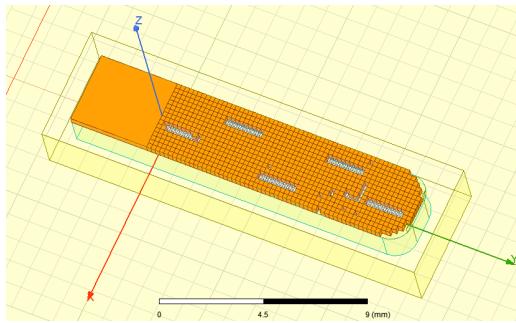


Figure 7.5: Final optimized antenna structure as simulated in ANSYS HFSS

The fundamental limitation of this approach stems from HFSS's computational complexity when handling numerous discrete geometric objects. Each cell in the grid requires individual creation, material assignment, and mesh generation within the electromagnetic simulation environment. For a typical discretization with dimensions $N_x = 48$ and $N_y = 100$, the solver must manage approximately 4,800 individual geometric entities.

This proliferation of objects creates several computational bottlenecks:

1. **Memory overhead:** Each cell requires separate data structures for geometry definition, boundary conditions, and material properties
2. **Meshing complexity:** The adaptive mesh generation algorithm must handle discontinuities at every cell boundary, significantly increasing the total number of mesh elements
3. **Matrix assembly:** The finite element method requires assembling system matrices that scale with the number of geometric interfaces

Consequently, simulations become computationally prohibitive as grid resolution increases, effectively limiting the achievable design precision and constraining the optimization search space.

7.3.2 Vectorized approach

The vectorized approach addresses the computational limitations of the grid method by implementing a unified geometric representation. Instead of creating individual cells, this method

generates contiguous regions of connected slots through geometric operations that merge adjacent cells with identical material properties.

The core algorithm processes the binary cell matrix to identify connected components of slot regions (*material* = -1) using flood-fill techniques. For each connected region, the method:

1. Extracts the region boundary using contour detection algorithms
2. Generates a single unified geometry object encompassing the entire slot region
3. Applies material properties to the consolidated object rather than individual cells

This vectorization dramatically reduces the number of geometric objects in the HFSS model, typically from thousands of individual cells to dozens of consolidated regions. The computational benefits include:

- Reduced memory requirements due to fewer geometric entities
- Simplified mesh generation with fewer discontinuous boundaries
- Faster matrix assembly and solution convergence
- Enhanced numerical stability in the electromagnetic simulation

7.4 Random-walk optimization

The antenna design optimization employs a probabilistic random-walk algorithm that iteratively evolves the slot configuration toward improved electromagnetic performance. The algorithm maintains a probability map that guides the modification of individual cells based on their contribution to the overall antenna performance.

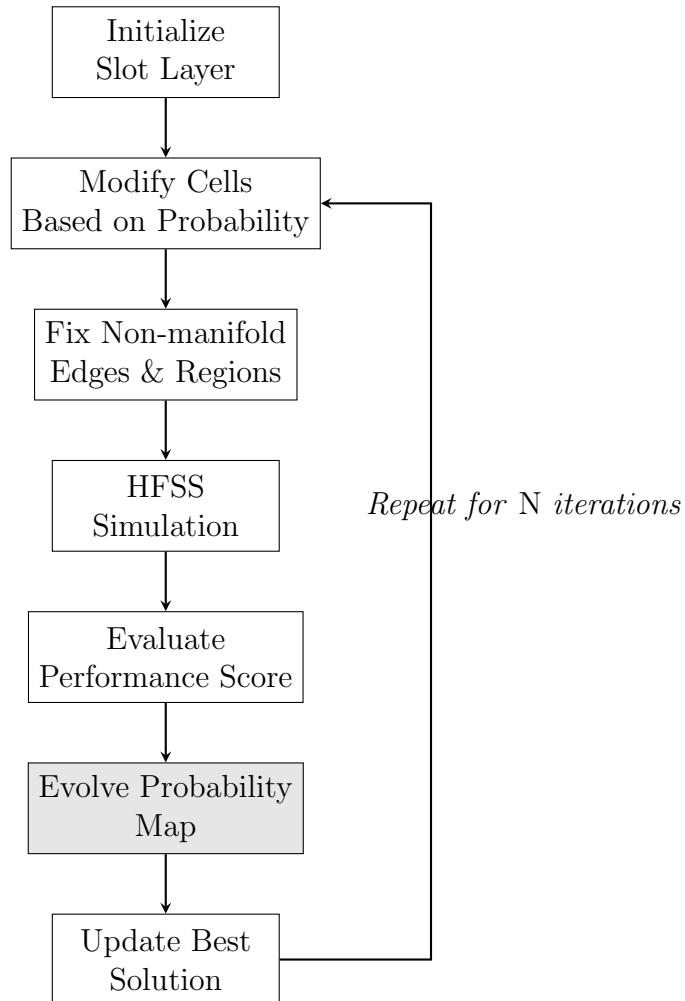


Figure 7.6: Random-walk optimization algorithm flow diagram

The optimization process operates through the following key mechanisms:

Probability-driven cell modification: Each cell (i, j) in the discretized antenna aperture maintains a change probability $p_{i,j}$ that determines the likelihood of material state modification (conductor \leftrightarrow air). The probability distribution evolves based on the performance feedback from electromagnetic simulations.

Multi-stage probability adaptation: The probability update mechanism operates through three distinct stages that collectively shape the exploration strategy:

Stage 1 - Direct modification feedback: For cells that were actually modified in the current iteration (i.e., cells where the material state differs from the local best solution), the probability

update follows:

$$p_{i,j}^{(k+1)} = \begin{cases} \max(\alpha_{win} \cdot p_{i,j}^{(k)}, p_{min}) & \text{if } \Delta S < 0 \text{ (improvement)} \\ \max(\alpha_{lose} \cdot p_{i,j}^{(k)}, p_{min}) & \text{if } \Delta S \geq 0 \text{ (degradation)} \end{cases} \quad (7.5)$$

where $\alpha_{win} < 1$ (reinforcement) and $\alpha_{lose} > 1$ (exploration) are learning rate parameters, and p_{min} is the minimum allowable probability.

Stage 2 - Gradient amplification: For each modified cell, the algorithm examines its four-connected neighbors and applies different probability updates based on both the performance outcome and the neighbor's material compatibility:

For neighbors sharing the same material as the modified cell:

$$p_{neighbor}^{(k+1)} = \begin{cases} \min(p_{neighbor}^{(k)} \cdot (2 - \alpha_{lose}), 1) & \text{if } \Delta S \geq 0 \text{ (reinforce similarity)} \\ \max(\alpha_{win} \cdot p_{neighbor}^{(k)}, p_{min}) & \text{if } \Delta S < 0 \text{ (reduce similarity)} \end{cases} \quad (7.6)$$

For neighbors with different material from the modified cell:

$$p_{neighbor}^{(k+1)} = \begin{cases} p_{explore} & \text{if } \Delta S < 0 \text{ (encourage boundary exploration)} \\ \max(\alpha_{lose} \cdot p_{neighbor}^{(k)}, p_{min}) & \text{if } \Delta S \geq 0 \text{ (discourage changes)} \end{cases} \quad (7.7)$$

where $p_{explore}$ is a high exploration probability value.

Stage 3 - Spatial diffusion: When a new local best solution is found, the algorithm applies a diffusion mechanism that smooths probability distributions across regions of similar material. For each cell, the probability is updated as:

$$p_{i,j}^{(k+1)} = (1 - \beta) \cdot p_{i,j}^{(k)} + \beta \cdot \bar{p}_{neighbors} \quad (7.8)$$

where β is the diffusion factor and $\bar{p}_{neighbors}$ is the average probability of neighboring cells that share the same material type in the local best solution.

Border enhancement: Cells located at material boundaries (conductor-air interfaces) receive enhanced modification probabilities, recognizing their critical role in determining electromagnetic field distributions and antenna characteristics.

The algorithm incorporates several regularization mechanisms to ensure physically realizable antenna geometries, including the elimination of non-manifold edges and the enforcement of minimum connected region sizes. These constraints maintain manufacturing compatibility while preserving the electromagnetic validity of the optimized designs. // ... existing code ...

7.5 Random Antenna Generation

As an alternative to the iterative optimization approach, the random antenna generation method provides a stochastic sampling strategy for exploring the antenna design space. This approach generates antenna configurations through probabilistic region growth algorithms, with the primary objective of creating a comprehensive dataset for neural network training.

7.5.1 Stochastic region generation

The random generation algorithm operates by creating multiple connected slot regions within the antenna aperture through a controlled growth process. Each antenna configuration is constructed by generating a random number of slot regions, where each region expands from a seed point according to probabilistic connectivity rules.

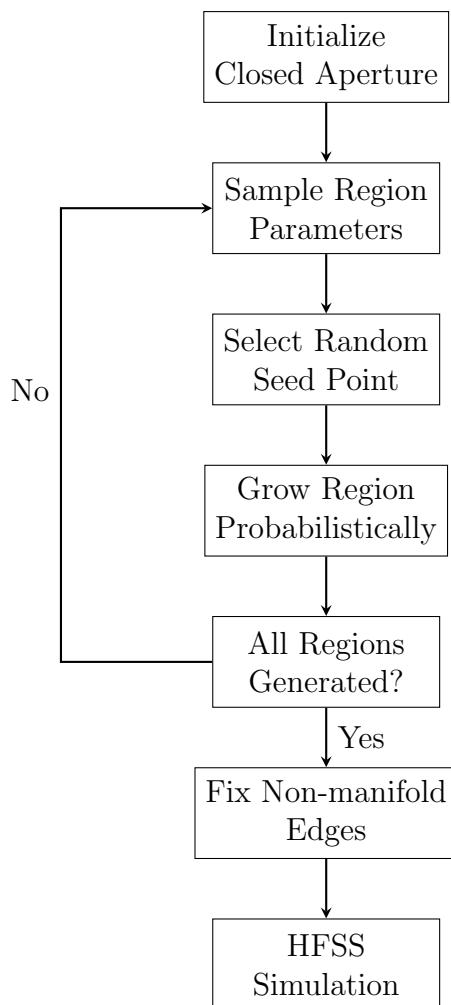


Figure 7.7: Random antenna generation algorithm flow

7.5.2 Region growth mechanism

The generation process begins with a fully closed antenna aperture (all cells set to conductor material). For each iteration, the algorithm:

Parameter sampling: The number of regions $N_{regions}$ is randomly sampled from a uniform distribution within predefined bounds $[N_{min}, N_{max}]$. For each region, the target size is sampled from a bimodal distribution that favors either small compact regions or larger extended regions:

$$N_{cells} = \begin{cases} \mathcal{N}(\mu_{small}, \sigma_{small}) & \text{with probability } p_{small} \\ \mathcal{N}(\mu_{large}, \sigma_{large}) & \text{with probability } (1 - p_{small}) \end{cases} \quad (7.9)$$

where $\mathcal{N}(\mu, \sigma)$ represents a normal distribution with mean μ and standard deviation σ .

Probabilistic connectivity: Each region grows from a randomly selected seed cell through iterative neighbor addition. At each growth step, a random cell from the current region boundary is selected, and its four-connected neighbors are evaluated for inclusion based on a connection probability p_{conn} :

$$p_{connection} = p_{conn} \cdot \gamma_{directional} \quad (7.10)$$

where $\gamma_{directional}$ is a directional bias factor sampled from a Rayleigh distribution, promoting elongated region shapes that are more conducive to electromagnetic wave propagation.

Growth termination: The region growth process continues until either the target number of cells is reached or a maximum iteration limit is exceeded, ensuring algorithm convergence even for highly constrained geometries.

This stochastic approach generates diverse antenna configurations with their corresponding electromagnetic performance metrics, creating input-output pairs suitable for supervised learning. The random generation provides systematic coverage of the design space, producing the large-scale datasets required for effective neural network training.

7.6 Discrete Metasurface Graph Neural Network

As an exploratory project, it has been proposed that the slot layer cover, where the slots are cut, could be produced using a Printed Circuit Board (PCB). As potential advantages, the possibility of drawing irregular shapes could allow for improved beam-forming performance and better and more reliable matching levels. On the other hand, the use of a dielectric in the PCB could distort the properties of the antenna and render the available theoretical models ineffective.

In this chapter, a novel approach to model waveguide antennas as irregular discrete metasurfaces of given resolution is proposed. The idea consists of using neural networks, and particularly,

Message Passing Neural Networks (MPNNs), to model slot interaction, matching levels, co-polarization, and cross-polarization radiation patterns. These models will incorporate direct physical relationships to ensure the correct behavior of the model.

The motivation for this strategy resides in the nature of the proposed antennas. Indeed, nearby slots exhibit neighbor-to-neighbor interactions, and, as will be analyzed later in this chapter, they can be effectively examined through *localized* admittance and geometric effects estimation. In other words, it is possible to disentangle the problem from global patterns by imposing well-informed physical constraints on the neural model.

7.7 Graph Definitions

In the context of metasurface antenna modeling, a graph $G = (V, E)$ is used to represent the physical structure of the antenna. Here, V is the set of nodes, each representing a slot in the waveguide structure, and E is the set of edges, representing the physical or electromagnetic interaction between pairs of slots.

Each node $v \in V$ is associated with a set of features h_v , which may include geometrical parameters and spatial location. Similarly, each edge $e_{vw} \in E$ is associated with a set of edge features e_{vw} that describe the type and strength of interaction between nodes v and w . These can include distance, relative orientation or region labeling in the case in which both nodes correspond to the same slot.

The graph is generally considered undirected (i.e., $e_{vw} = e_{wv}$) unless directional effects are explicitly relevant to the model. The neighborhood of a node v , denoted $\mathcal{N}(v)$, consists of all nodes w such that $(v, w) \in E$.

This graph-based abstraction allows us to compactly represent localized interactions between slots while preserving the spatial and physical structure of the antenna, which is essential for accurate modeling.

7.8 KNN-based pooling

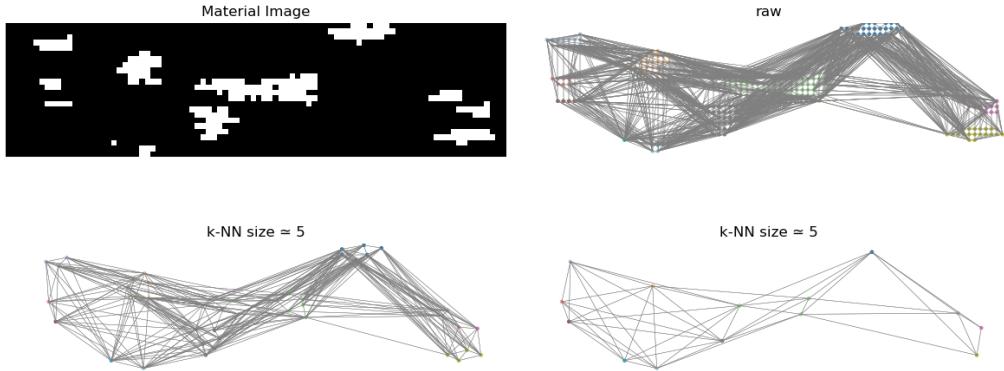


Figure 7.8: 2-stage KNN graph pooling with a ratio 5:1 / Same regions are represented by the same color.

In this section, we describe a graph pooling strategy based on the k -Nearest Neighbors (KNN) algorithm that enables the reduction of graph complexity while preserving the spatial and topological information of slot arrangements in metasurface antenna design.

The core motivation for KNN-based pooling lies in the necessity to aggregate spatially close nodes (i.e., slots) into coarser representations, while preserving essential electromagnetic interactions and localized physical relationships. To achieve this, a region-based KNN pooling strategy is proposed, structured into the following steps: In Figure 7.8, an example on how this pooling is performed is given.

Region Detection via Depth-First Search (DFS)

The graph representing the metasurface antenna is first partitioned into connected regions. Each region consists of a group of slots that are directly or indirectly connected. This partitioning is achieved using a Depth-First Search (DFS) algorithm, which recursively explores all neighboring nodes from a given starting node.

Given a graph $G = (V, E)$, the DFS algorithm proceeds as follows:

1. Initialize all nodes in V as unvisited.
2. Select an unvisited node $v \in V$ and start a DFS traversal from v .
3. During traversal, mark each visited node and record its membership in the current region.
4. Recursively visit all unvisited neighbors $w \in \mathcal{N}(v)$.
5. Repeat the process for all remaining unvisited nodes to identify all regions.

Each discovered region is then considered a subgraph, over which localized pooling is performed.

Localized KNN-Based Pooling

Once regions are identified, a KNN-based pooling operation is applied within each region. This localized pooling approach ensures that spatial positions and localized interactions are maintained.

For each node v within a region:

1. Compute the Euclidean distance between v and all other nodes in the same region based on their spatial coordinates.
2. Identify the k nearest neighbors of v , denoted $\text{KNN}_k(v)$.
3. Aggregate the features of v and its k neighbors, typically by computing the mean.
4. Replace v and its neighbors with a single pooled node with features h_{pool} .

This operation effectively reduces the number of nodes while preserving feature smoothness and local slot arrangement.

Edge Preservation and Region Boundary Encoding

During pooling, it is critical to preserve the structural information encoded in the graph edges. Therefore, edge relationships are updated to reflect new connections between pooled nodes. Additionally, metadata is introduced on each edge to indicate whether the connected nodes belong to the same original region or span across region boundaries.

Let e_{vw} be an edge between two pooled nodes v and w . An auxiliary edge feature r_{vw} is defined as:

This enables the neural architecture to distinguish between intra-region and inter-region interactions, which can be critical for modeling localized vs. global coupling effects.

By combining region-based graph decomposition with KNN pooling, this approach offers an efficient and physically interpretable method to reduce graph dimensionality while maintaining the integrity of spatial and topological relationships among the slots.

7.9 Message Passing Neural Networks (MPNNs)

Graph Neural Networks (GNNs) is a term widely used to refer to neural architectures that process graphs. MPNNs are a specific type of GNN that establish a powerful framework. While originally used in quantum chemistry to predict molecular structure properties, these architectures today show great potential in a vast range of domains, such as computer vision, communication networks, and social network interactions. In this text, a physics-enhanced approach is proposed to integrate MPNNs into metasurface antenna design.

The most general theoretical framework, as described in [Gilmer et al., 2017], establishes the architecture to obtain the next-stage hidden features of node v in three steps: the first is the message construction at stage t , $M_t()$, which is usually defined as a Multi-Layer Perceptron (MLP) operating on the hidden node features h_v^t and h_w^t and the edge features e_{vw} for all connected neighbors w of v . The second step consists of applying an aggregation function to these messages for node v , such as a global sum or mean function. The third step consists of an update function $U_t()$, which is usually also an MLP, applied to the output of the second step. Mathematically, this translates to the following relationships:

$$\begin{aligned} m_v^{t+1} &= \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}) \\ h_v^{t+1} &= U_t(h_v^t, m_v^{t+1}) \end{aligned} \tag{7.11}$$

Several elements can be added to improve the model:

- Residual layers: Adding the input to the output after each layer enable the optimizer to skip layers when they probe unnecessary.
- PReLU activation functions: They have been proven to be efficient in GNNs to increase model depth.

7.10 Auto-Encoder as a geometric descriptor

Autoencoders provide a powerful framework for learning compact representations of high-dimensional data, particularly in contexts where spatial or structural coherence is important. In regression-based tasks, they can be used to derive localized feature descriptors that map an input space to a corresponding output space. By training an encoder-decoder architecture to reproduce the input image as its output, the model is encouraged to learn a compressed, informative representation at the bottleneck layer. This latent space captures essential geometric features while discarding redundant information, which helps in reducing overfitting. In Figure 7.9 the simplified architecture is presented.

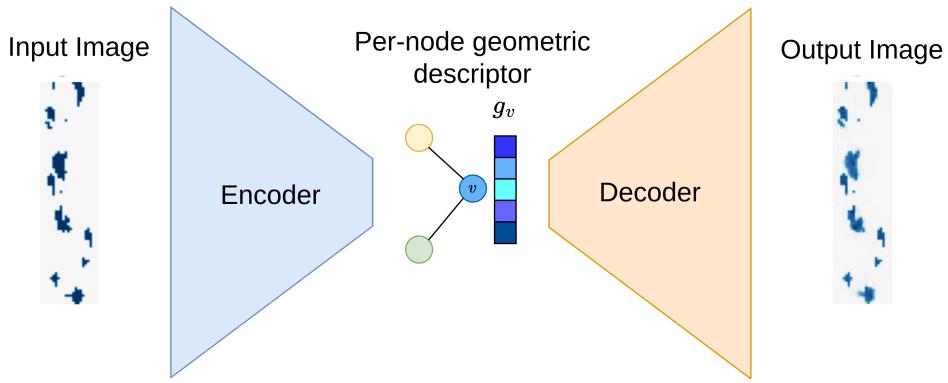


Figure 7.9: Auto-Encoder architecture to extract geometric descriptors

Extending this principle to graph-structured data allows the model to learn rich, spatially grounded embeddings at the node level. Each node in the graph represents a localized image patch or feature region, and the message-passing mechanism captures relationships among neighboring nodes. As a result, the latent features learned at each node encode a local geometric context, enabling the model to infer or compare complex structural patterns.

Learning node-level descriptors that faithfully represent local slot geometries supports accurate downstream predictions and enhances the interpretability of the model's internal representations. Additionally, the auto-encoder is not limited by data availability, since new samples can be generated on-the-run, therefore allowing for a more precise encoding than if using the computationally expensive slot antenna dataset.

7.10.1 Encoder

The encoder architecture is designed to extract a low-dimensional geometric descriptor from image data by leveraging both convolutional and graph-based operations. It begins with a convolutional neural network (CNN) that processes the input binary image to generate a dense spatial feature map. This map is sampled at node positions defined by the input graph, yielding per-node feature vectors. These vectors are then passed through a stack of message-passing neural network (MPNN) layers, which perform neighborhood aggregation based on the edge structure of the graph. Local self-interaction layers, implemented as multilayer perceptrons (MLPs), refine node representations after message passing.

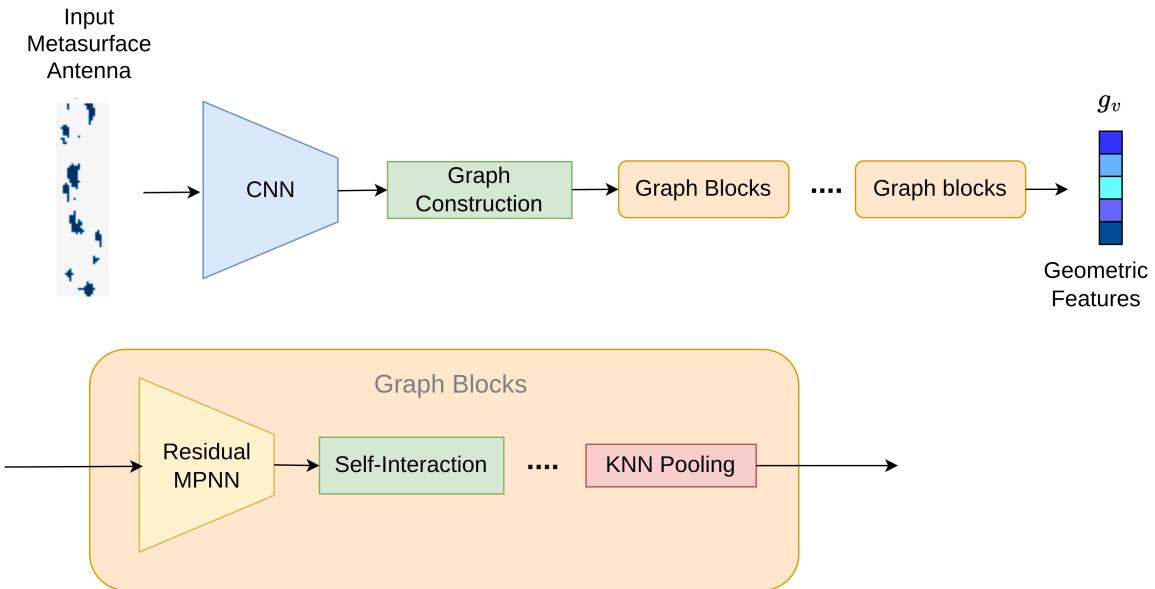


Figure 7.10: Encoder detailed Architecture

To incorporate hierarchical abstraction, the encoder applies K-nearest neighbor (KNN) pooling at specified stages. For each pooling stage, nodes are clustered, and their features are aggregated using mean pooling. New graphs are constructed for the pooled nodes using precomputed edges and positions. This progressively reduces the number of nodes while enriching feature representations. The final node features are then projected into a fixed-size latent vector $z \in \mathbb{R}^d$ through a fully connected layer, serving as the compact geometric descriptor of the input graph.

7.10.2 Decoder

The decoder reconstructs the spatial image from the latent representation by reversing the encoder’s operations. Starting from the latent vector z , a fully connected layer expands it to initialize node features for the coarsest graph structure used during the last pooling stage. These features are refined through a series of MPNN layers and corresponding self-interaction layers, which propagate information across the graph and enhance local detail.

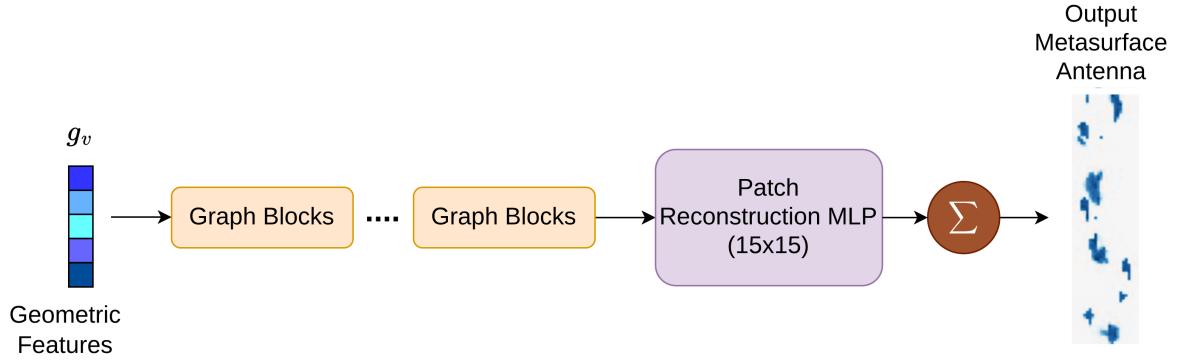


Figure 7.11: Encoder detailed Architecture

Subsequently, each node's final feature vector is passed through a patch-generating linear layer to yield a small image patch of size $a \times b$. These patches are placed back into the output image space using the original node positions. The reconstructed image \hat{x} is obtained by summing the overlapping patches. This patch-wise reconstruction enables spatially informed generation while preserving the underlying graph-based representation.

7.10.3 Training Objective

To train the autoencoder, a mean squared error (MSE) loss is computed between the input image x and the reconstructed output \hat{x} :

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2.$$

Minimizing this loss ensures that the reconstructed image closely approximates the original input. Because the reconstruction depends entirely on the latent representation z , accurate recovery implies that z contains sufficient information about the geometric structure and visual content. Consequently, the autoencoder learns an effective geometric descriptor that captures local and spatial features of the graph-structured input.

During training, we employed a synthetic dataset whose samples were drawn from a distribution slightly more complex than that of the validation set, featuring additional and larger regions. To enhance model robustness, we incrementally increased the dataset size: whenever the validation loss failed to improve below its previous minimum within the specified patience window, we doubled the number of synthetic samples.

7.10.4 Hyperparameter choice

Table 7.1: Hyperparameter settings for synthetic data generation, training, and model architecture.

Parameter	Value
<i>Dataset Split</i>	
Train samples	Real-time x2 augmentation (up to 8096 samples)
Val samples	128
<i>Training</i>	
Learning rate	1×10^{-4}
Batch size	16
Max epochs	10 000
Early-stop patience	500
<i>LR Scheduler</i>	
Reduction factor	0.5
Patience	50
Threshold	1×10^{-4}
Min. LR	1×10^{-6}
<i>Model Architecture</i>	
CNN channels	[256, 256, 128, 128, 128, 64, 64, 64]
Layers per stage	[1, 1, 1]
Hidden dims per stage	[64]; [32]; [16]
Decoder hidden dims	[32, 64, 128, 256]
Output SIB dims	[512, 1024, 1024, 2048]
Pooling sizes (k)	[6, 6]
Latent dimension	12 or 32
Patch size (h×w)	15×15

NOTE: SIB refers to Self-Interaction Blocks, Hidden dimensions per stage refer to the dimensions of MPNN layers.

7.10.5 Results

Figure ?? shows the evolution of the training loss alongside the schedule of synthetic sample counts. Notice that each time we double the size of the synthetic dataset, there is a temporary uptick in the training loss, which subsequently enables the model to achieve lower validation losses. Figure ?? directly compares the validation loss before and after each augmentation step, highlighting the benefit of our incremental data-expansion strategy.

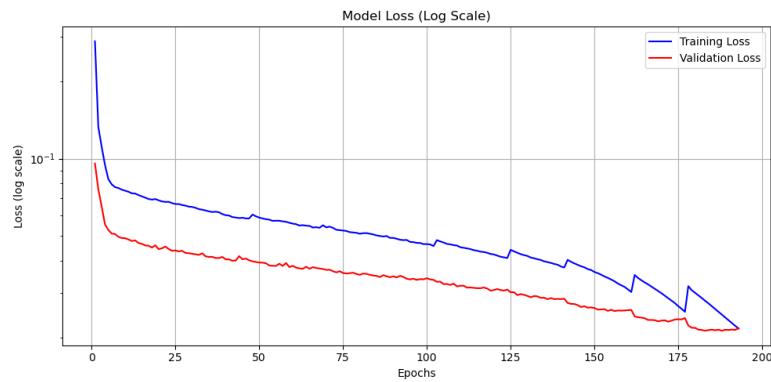


Figure 7.12: Enter Caption

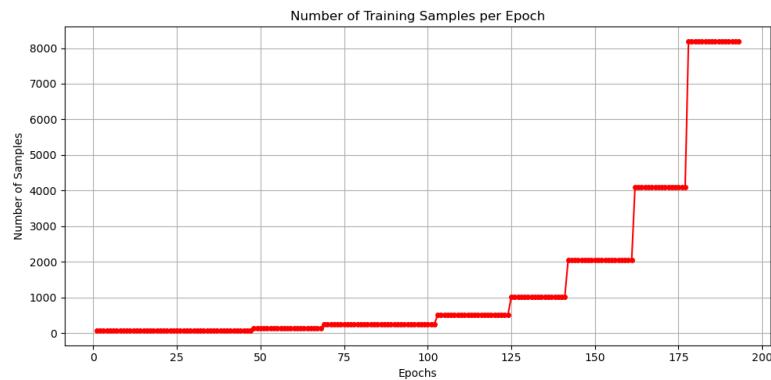


Figure 7.13: Enter Caption

Reconstruction results on a representative subset of the synthetic training data are presented in Figure 7.14, while Figure 7.15 illustrates model predictions on unseen validation samples. In both cases, the autoencoder faithfully captures the underlying structure of the input, with only minor deviations.

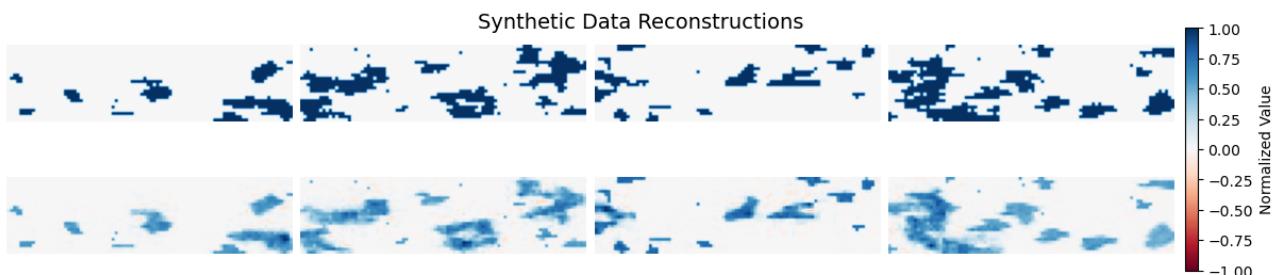


Figure 7.14: Enter Caption

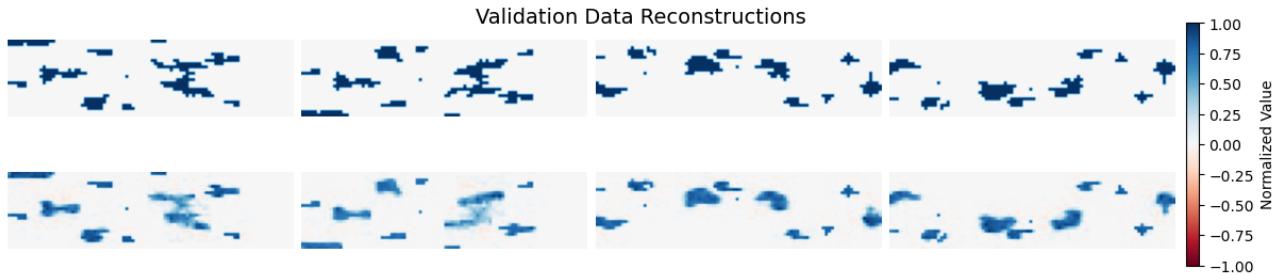


Figure 7.15: Enter Caption

A robustness assessment of the node-wise geometric embeddings was conducted by injecting additive random noise at varying amplitudes and then examining the decoder’s reconstruction fidelity (see Figure 7.16). The reconstructions degrade gradually with increasing noise intensity, demonstrating a smoothly varying response rather than abrupt failure. This latent-space smoothness is a desirable property for downstream tasks, as it suggests the autoencoder has learned a stable, well-structured manifold that will facilitate reliable training in subsequent models.

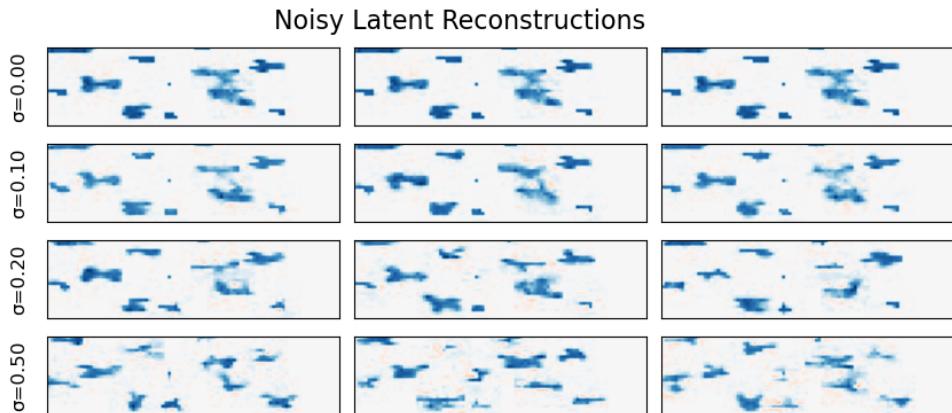


Figure 7.16: Enter Caption

7.11 Architecture

7.11.1 Regularization

LayerNormalization: A powerful regularizer to ensure meaningful representations are obtained.
Weight decay

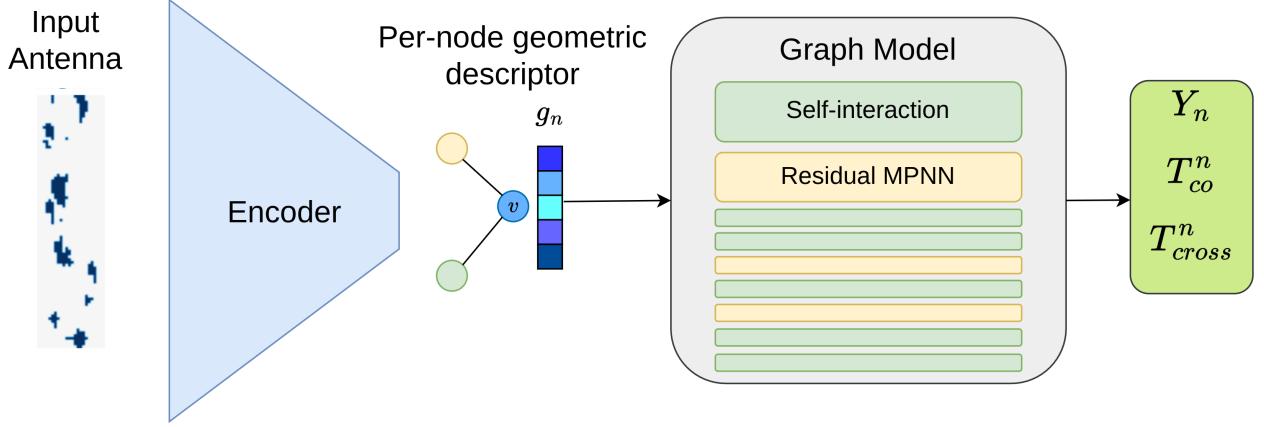


Figure 7.17: Enter Caption

7.11.2 Positional Encoding

7.12 SWA Metasurface Theoretical Model

In this section we develop an approximative theoretical foundation underlying the SWA metasurface synthesis, casting the problem in terms of transmission-line theory augmented by shunt admittances that represent the individual slots. By expressing each slot and inter-slot segment as an ABCD matrix, we obtain closed-form expressions for global input admittance, local voltages, and ultimately the radiated field pattern.

The power of this representation yields on the fact that only local elements to each slot will need be predicted by the neural network. This should allow the network to converge to a correct solution.

7.12.1 Slot Voltage Prediction

In Equation xxx, a relationship is given between the normalized admittance, the waveguide voltage, the slot voltage and the geometry of each slot. If slots are not rectangular, this relationship will probably not hold, so an approximation is proposed hereafter:

$$V_n^s = T(g_v)V_n Y_n^a,$$

where g_n is the output of node n from the encoder, and T is a function that aims to gather the geometric factor in Elliot's classical equation. Since irregular slots can have a dual polarization, different T functions will be used for co- and cross-polarization, T_{co} and T_{cross} . For co-polarisation, a sign will also be forced upon $T(g_v)$, in order to account for the phase mirroring effect due to surface currents in the broad face of the waveguide, as depicted in Figure xx.

7.12.2 Radiation Pattern Predictions

Finally, each slot acts as a localized radiator with complex excitation V_n^s . Denoting the slot position vector by \mathbf{r}_n and orienting the aperture in free space, the array-factor contribution is

$$E(\theta, \phi) \propto \sum_{n=1}^N V_n^s e^{j k \hat{\mathbf{r}} \cdot \mathbf{r}_n},$$

where $\hat{\mathbf{r}} = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$ and $k = 2\pi/\lambda$. Squaring the magnitude yields the power pattern. A final multiplication by the ground-plane reflection pattern, which is defined by learnable phi and theta projections, modulates $E(\theta, \phi)$ to account for mirror currents, completing the full-space radiation prediction.

This complete physics pipeline is described in Figure ??

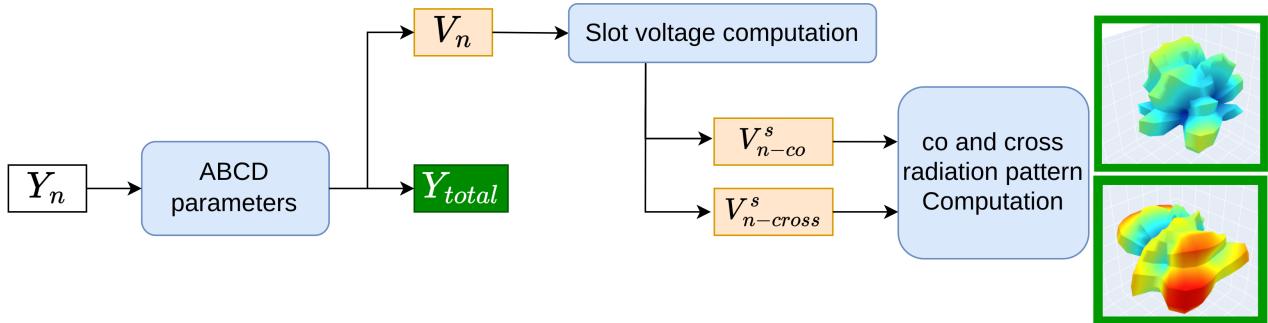


Figure 7.18: Enter Caption

7.13 Earth Moving Distance (EMD) for radiation patterns

7.13.1 Motivation

When comparing predicted and target radiation patterns on the sphere, a direct point-wise loss such as Mean Squared Error (MSE) can be problematic. Radiation patterns often exhibit sharp, localized peaks (lobes), and small angular shifts in these peaks can lead to disproportionately large MSE values, even if the overall distribution shape is preserved. This is because MSE penalizes discrepancies in amplitude at corresponding fixed grid locations, and does not account for the geometric shift or spread of radiation energy. Consequently, even a minor misalignment of lobes can destabilize training, leading to poor convergence or collapsed predictions. To address this, the Earth Mover's Distance (EMD), also known as the Wasserstein distance, offers a distributional comparison. It measures how much 'effort' it takes to morph the predicted

pattern into the target, taking spatial geometry into account, and thus offering a more robust and physically grounded comparison.

7.13.2 Definition

EMD is defined as the minimal cost required to transform one distribution into another, where the cost is determined by the distance over which probability mass must be transported. In the context of radiation patterns defined over the sphere, the input and target are treated as discrete distributions over spherical coordinates , with each location carrying a certain amount of energy. The cost of moving energy from one point to another is computed based on a toroidal metric, which maps angular coordinates to 3D unit vectors and uses Euclidean distance in 3D space.

7.13.3 Sinkhorn approximation

Computing exact EMD is computationally expensive, especially over dense spherical grids. To make this tractable, we use the Sinkhorn approximation, which regularizes the transport problem with an entropic penalty, enabling efficient iterative solution via the Sinkhorn algorithm. This approximation allows for GPU acceleration and differentiability, which are critical for integration into neural network training loops. In practice, this is implemented through the GeomLoss library's SamplesLoss with loss=`"sinkhorn"`.

7.13.4 GeomLoss - hyperparameter choice

The behavior and smoothness of the Sinkhorn-approximated EMD are governed by several hyperparameters:

- **blur: 0.0001** Controls the strength of the entropic regularization. Smaller values approach the true EMD but require more computation and can be less stable.
- **scaling: 0.9** Controls the decay of the convolution kernel in the Sinkhorn loop. A value close to 1 ensures fine control, improving convergence.
- **diameter: 2** The maximum possible cost/distance in the space. This should reflect the maximum spatial extent over the sphere.
- **reach: 1** Influences the neighborhood size used in the convolution kernel for computing the cost; higher values can increase stability.

These parameters must be carefully chosen to balance accuracy, numerical stability, and computational efficiency in training deep models for radiation pattern prediction.

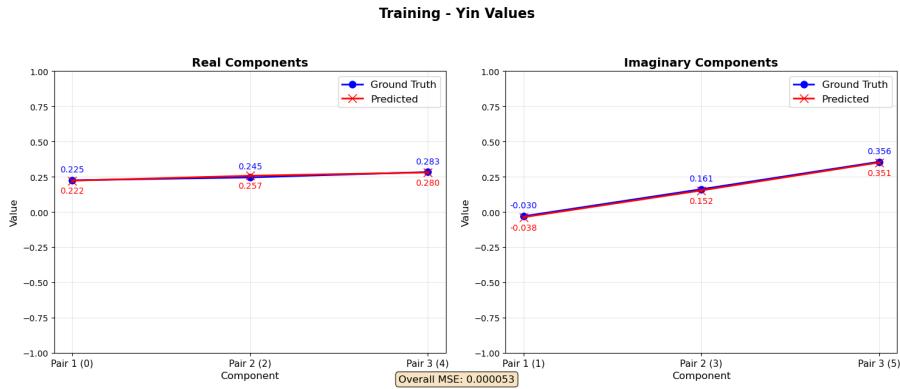
7.14 Training

7.14.1 Overfitting Test

Testing whether a complex model can overfit is an essential step in validating its capacity to learn and converge. Demonstrating overfitting confirms that the model architecture is expressive enough to fit the training data, which is a prerequisite for generalization when trained on larger datasets.

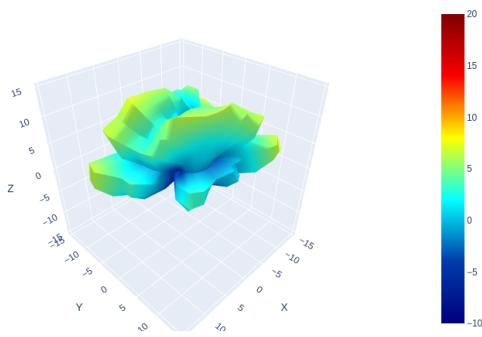
To assess the model's ability to converge to the correct solution given sufficient training time and computational resources, we conduct an overfitting experiment using a deliberately small dataset of just 256 simulations. The model is trained for 500 epochs on this dataset. The intent is not to achieve generalization here, but to verify that the model can minimize training error effectively.

The results of this overfitting experiment are presented in Figures 7.24, which illustrate the model's performance on the training set. A good overfitting is obtained for all predictions.



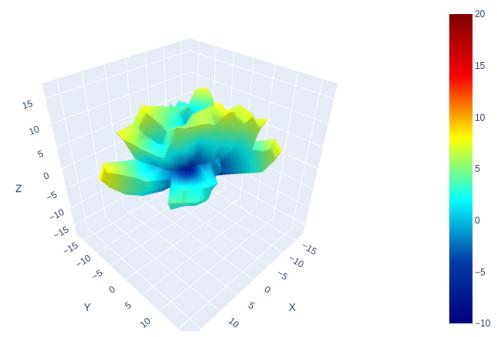
(a) Input Yin

Training - Predicted co-polarisation



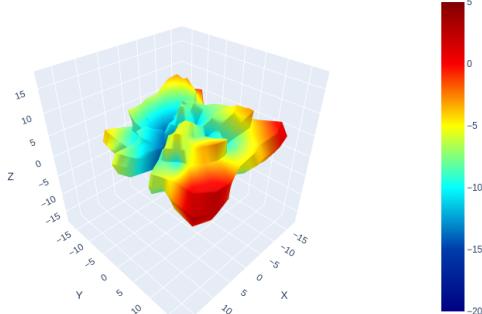
(a) Co prediction

Training - Ground Truth co-polarisation



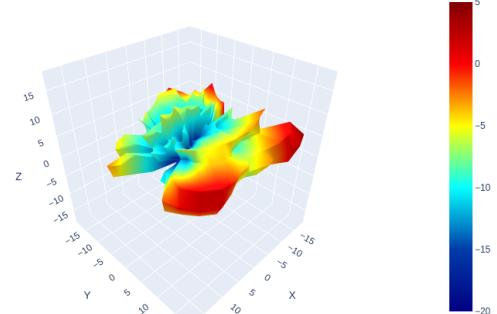
(b) Co ground truth

Training - Predicted Cross-polarisation



(c) Cross prediction

Training - Ground Truth Cross-polarisation



(d) Cross ground truth

Figure 7.19: Top: input Yin. Middle: co-prediction vs ground truth. Bottom: cross-prediction vs ground truth.

7.14.2 Hyperparameter choice

Table 7.2: Hyperparameter settings for data handling, training, and model architecture.

Parameter	Value
<i>Dataset Configuration</i>	
Total samples	3 930
Train split	7 600
Validation split	256
<i>Graph Construction</i>	
Max distance	[20, 14] (cells in x and y)
Max offset	2 (open-cell steps)
<i>Training</i>	
Batch size	32
Learning rate	1×10^{-4}
Weight decay	1×10^{-6}
Dropout probability	0.1
Max epochs	1 000
<i>Optimizer (Adam)</i>	
β_1	0.7
β_2	0.9
<i>Gradient Clipping</i>	
Max gradient norm	1.0
<i>LR Scheduler</i>	
Enabled	True
Initial LR	1×10^{-4}
Target LR	1×10^{-6}
Decay start epoch	200
Decay duration (epochs)	800
<i>Loss Weights</i>	
$\lambda_{\text{EMD_gain}}$	0.001
$\lambda_{\text{EMD_cross}}$	0.002
λ_Y	5
<i>Model Architecture</i>	
Hidden dimension	32
Layer sequence ¹	Alternating SI-MPNN
Yin hidden dimension	32
Geometric factor hidden dim.	64

¹ 7 layers total: SI-MPNN-SI-MPNN-SI-MPNN-SI.

7.14.3 Results

Loss Evolution

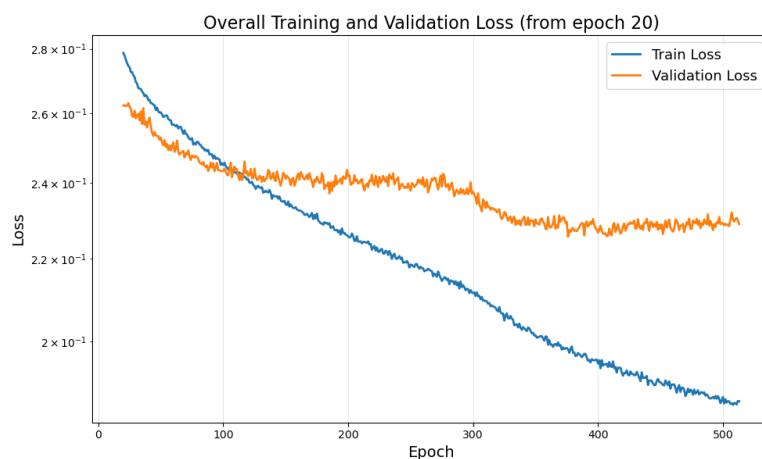
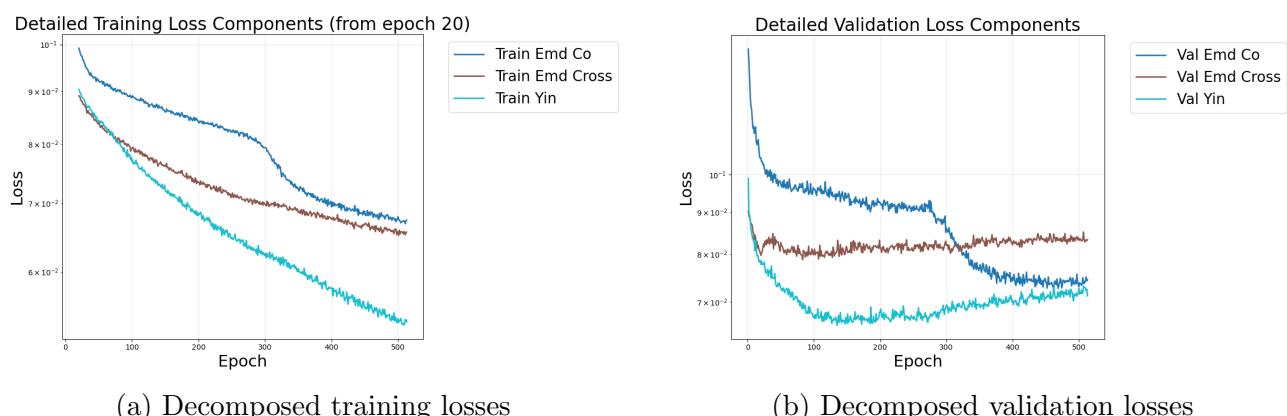


Figure 7.20: Training and validation losses



(a) Decomposed training losses

(b) Decomposed validation losses

Figure 7.21: Comparison of decomposed training versus validation losses

Loss Histograms

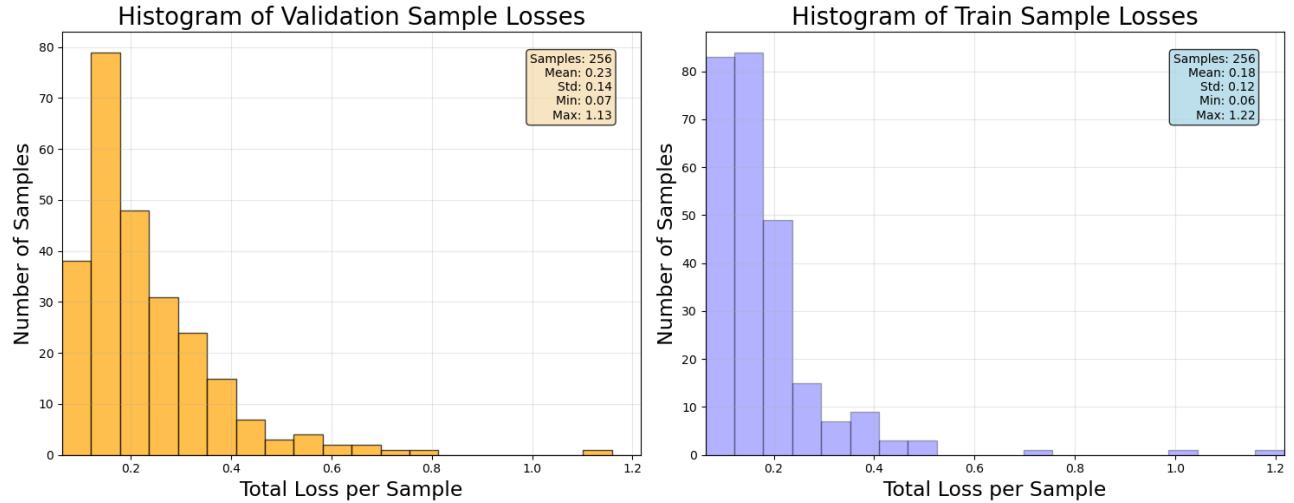


Figure 7.22: Loss Histograms

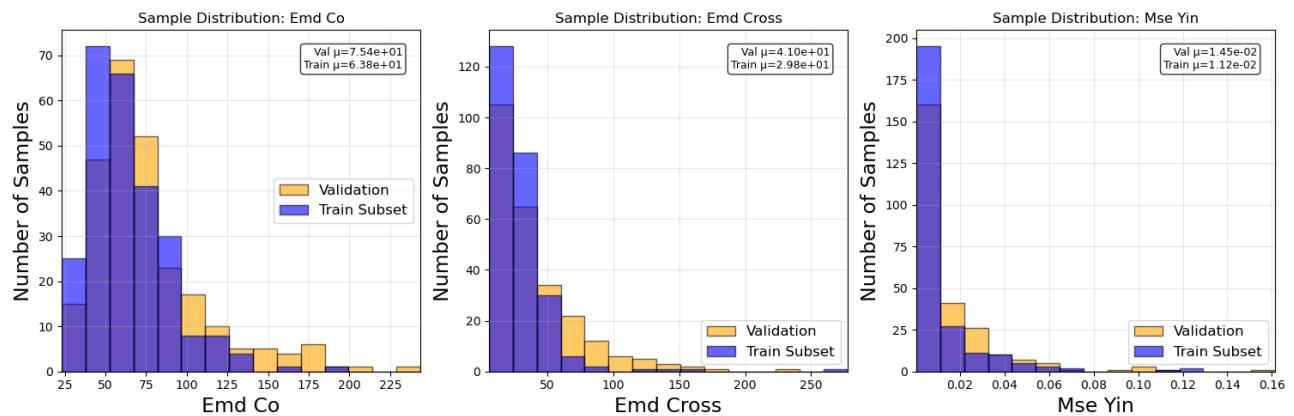
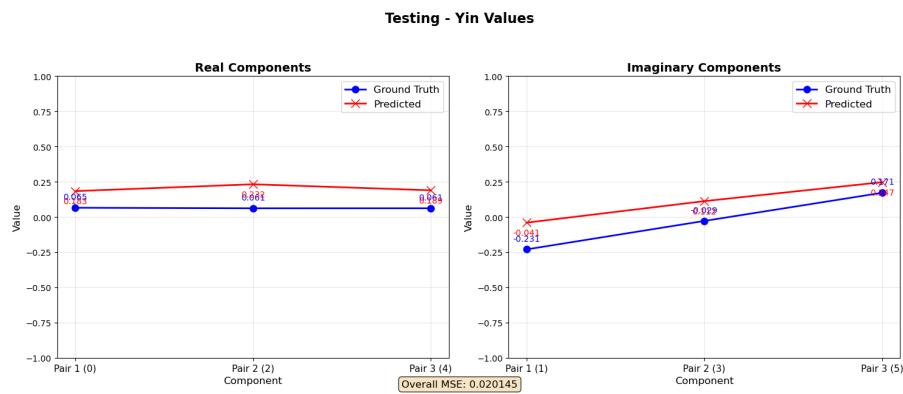


Figure 7.23: Detailed Loss histograms

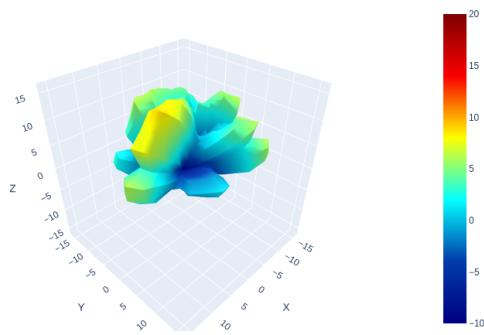
Validation Example

A typical ground truth and predictions for a validation sample look as shown in Figure ??.



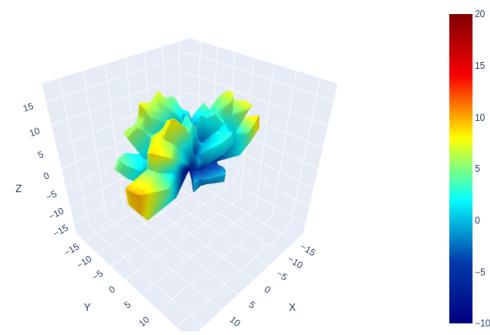
(a) Input Yin

Testing - Predicted Gain



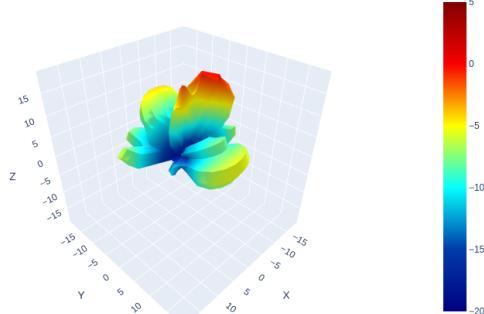
(a) Co prediction

Testing - Ground Truth Gain



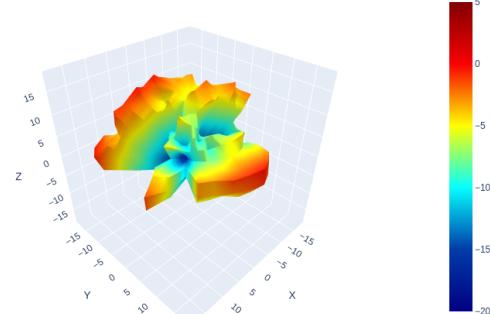
(b) Co ground truth

Testing - Predicted Cross



(c) Cross prediction

Testing - Ground Truth Cross



(d) Cross ground truth

Figure 7.24: Top: input Yin. Middle: co-prediction vs ground truth. Bottom: cross-prediction vs ground truth.

Model analysis

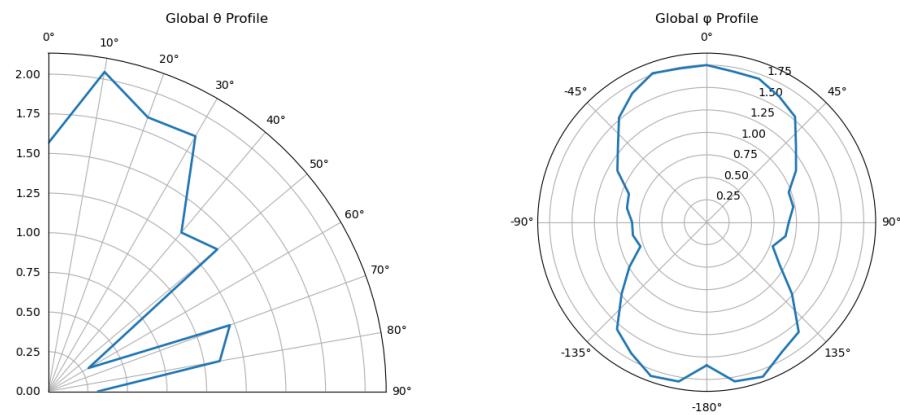


Figure 7.25: Loss Histograms

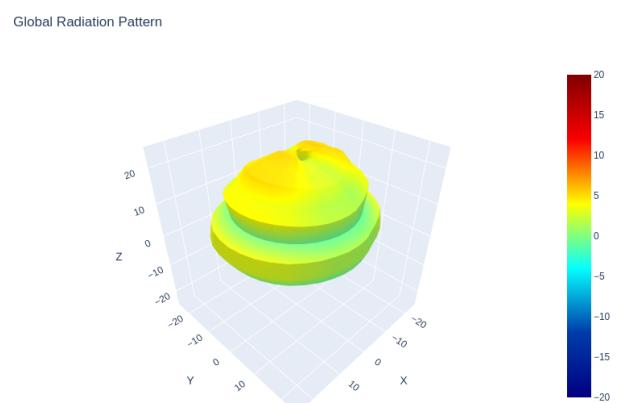


Figure 7.26: Loss Histograms

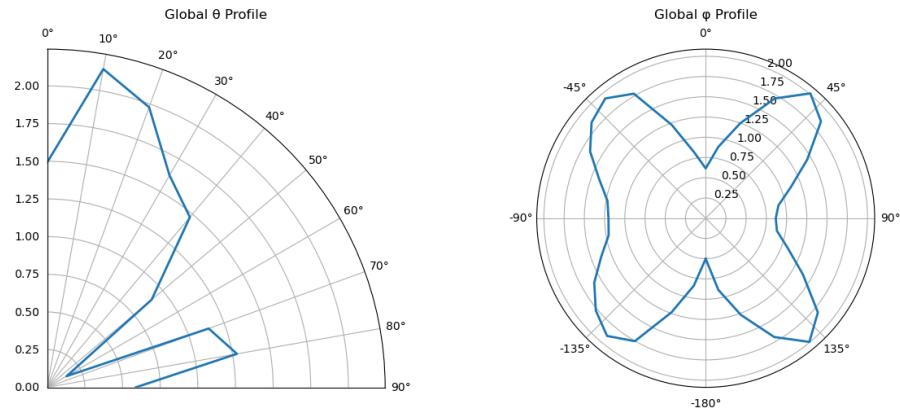


Figure 7.27: Loss Histograms

Global Radiation Pattern

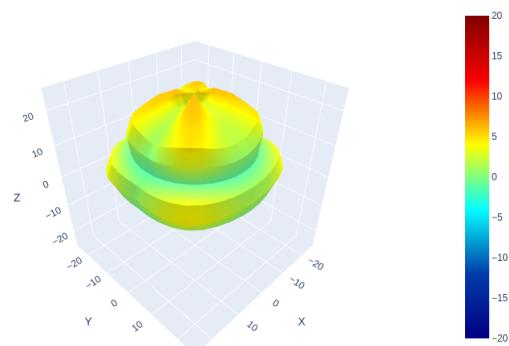


Figure 7.28: Loss Histograms

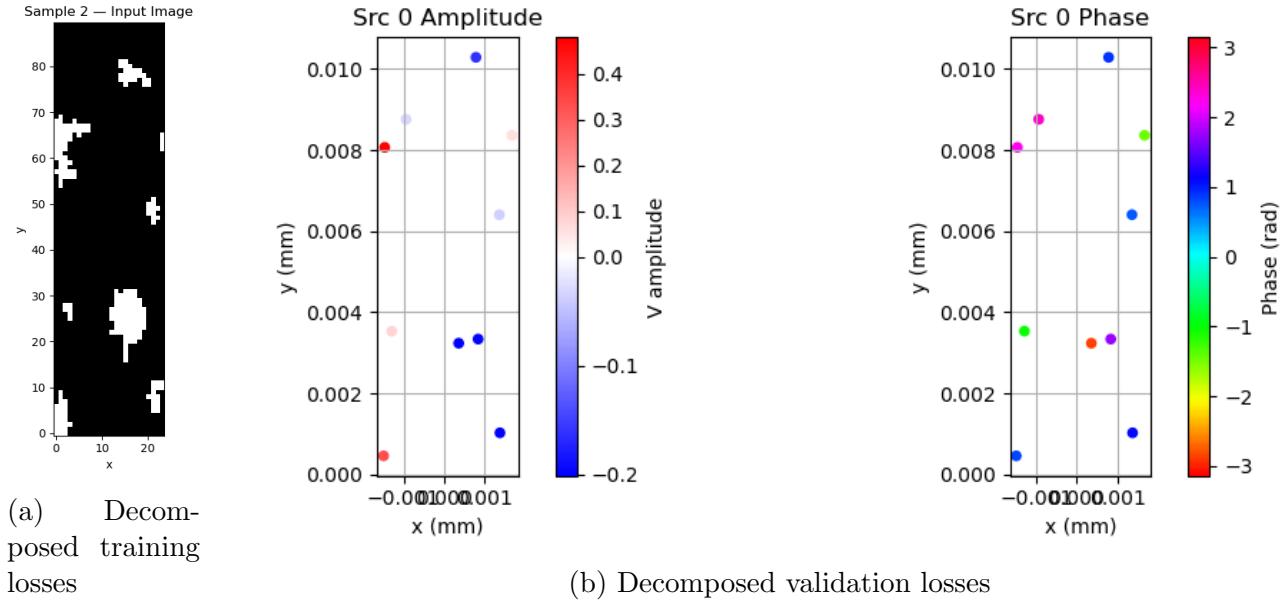


Figure 7.29: Comparison of decomposed training versus validation losses

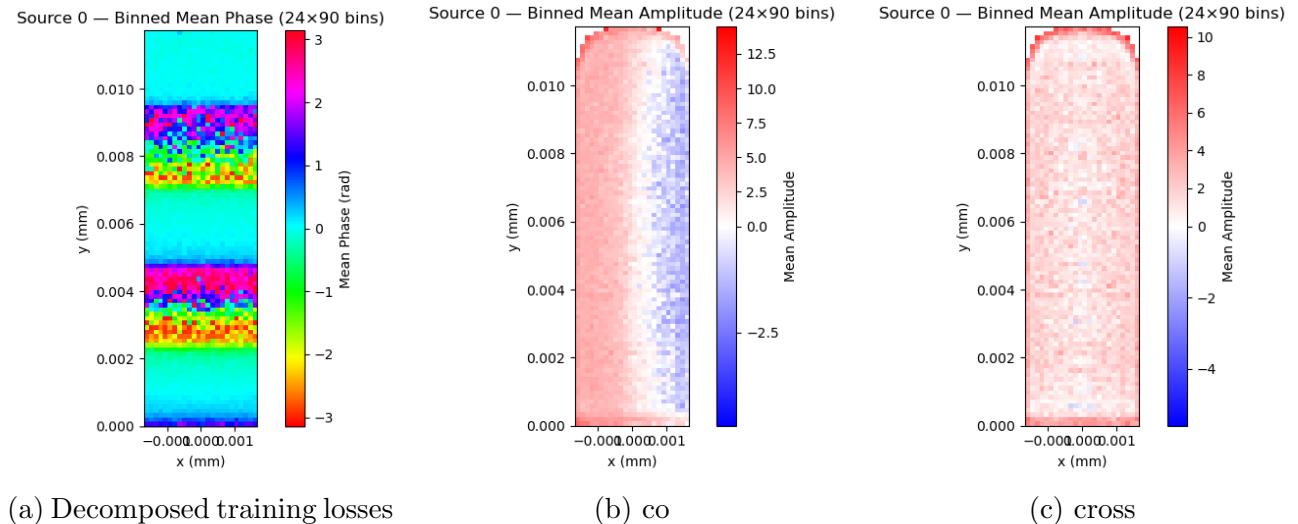


Figure 7.30: Comparison of decomposed training versus validation losses

Chapter 8

Conclusion

Bibliography

[Gilmer et al., 2017] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212.