REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF AVIATION ELECTRICS AND ELECTRONICS

GRADUATION THESIS

ARTIFICIAL INTELLIGENCE ASSISTED SURROGATE MODELING AND OPTIMIZATION OF HELIX ANTENNA DESIGN FOR SATELLITE MISSIONS

Advisor: Assoc. Prof. Peyman MAHOUTI

190B1016 – Uygar Tolga KARA

İstanbul, 2023

T.C.

YILDIZ TECHNICAL UNIVERSITY

DEPARTMENT OF AVIATION ELECTRICS AND ELECTRONICS

ARTIFICIAL INTELLIGENCE ASSISTED SURROGATE MODELING AND OPTIMIZATION OF HELIX ANTENNA DESIGN FOR SATELLITE MISSIONS

A graduation thesis submitted by Uygar Tolga KARA in partial fulfillment of the requirements for the degree of bachelor is approved by the committee on 15.06.2023 in Department of Aviation Electrics and Electronics, Faculty of Applied Sciences.

Examining Committee

Assoc. Prof. Ufuk SAKARYA
Yildiz Technical University ______________________

Assoc. Prof. Peyman MAHOUTİ
Yildiz Technical University ______________________

Asst. Prof. Mumin Tolga EMİRLER
Yildiz Technical University ______________________

This project is supported by TÜBİTAK BİDEB ☐2209A / ☐2209B.

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my graduation thesis under the title of Artificial Intelligence Assisted Surrogate Modeling and Optimization of Helix Antenna Design for Satellite Missions supervised by my advisor, Assoc. Prof. Peyman MAHOUTI. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Uygar Tolga KARA

Signature

PREFACE

I would like to express my heartfelt gratitude to my family. They supported me from the day I started university and have not stopped supporting me until now. I am grateful for their support, understanding and love. My mother especially kept reminding me of my exam times and the start of hours of classes. I hope my graduation makes them happy.

I also want to express my sincere gratitude for my 2 best friends since high school. I have them to thank for helping me to keep my mental health. They kept fun company.

I also would like to express my sincere gratitude to my teachers in my faculty. Over the years, they always tried to give more information to us, did not get tired of answering our questions and tried to prepare us for the jobs. Also, I want to thank them for their friendly attitude. While mentioning our teachers, I would also like to thank Yildiz Technical University as a whole. Starting my second degree in this university gave me hope, helped me to understand the job I like to do and helped me to get into the job. Also, I am amazed at productivity and teamwork inside voluntary student clubs.

Lastly, I would like to thank my advisor, Assoc. Prof. Peyman MAHOUTI. Since our teacher came to school, I had fun listening to his lessons. I think his conversational, and informative method of teaching, and the relationship of his classes with modern technologies did help to make his lessons more fun to follow. In fact, my negative opinion about microwave related topics changed completely and as a result I wanted to write my graduation thesis based on modeling of antennas with deep learning.

This graduation thesis consists of design of optimization of helix antennas with the help of deep learning algorithms. Currently, technology advancement has increased exponentially again since the release of AI tools and chatbots. In the near future, as a guess much more efficient methods may be used for designing antennas. We hope that this thesis can give insights on helix antenna design and optimization for researchers.

June, 2023

Uygar Tolga KARA

ÖNSÖZ

Aileme içtenlikle teşekkür etmek istiyorum. Üniversiteye başladığım günden bu yana bana destek oldular ve desteklerini hala esirgemiyorlar. Destekleri, anlayışları ve sevgileri için minnettarım. Özellikle annem, sınav saatlerimi ve online derslerin başlama saatlerini bana sürekli hatırlattı. Umarım, hayatları boyunca mutlu olurlar.

Lise yıllarımdan beri 2 en iyi arkadaşıma da içtenlikle teşekkür etmek istiyorum. Devam eden arkadaşlıkları için onlara minnettarım. Onlarla vakit geçirmek hep eğlenceliydi.

Fakültemdeki öğretmenlerime de içtenlikle teşekkürlerimi sunmak istiyorum. Yıllar boyunca bize daha fazla bilgi vermeye çalıştılar, sorularımızı yanıtlamaktan yorulmadılar ve bizi işe girmemiz için hazırlamaya çalıştılar. Öğretmenlerimizden bahsederken, Yıldız Teknik Üniversitesi'ne genel olarak da teşekkür etmek isterim. Bu üniversitede ikinci dereceme başlamak, bana umut verdi, yapmayı sevdiğim işi anlamama yardımcı oldu ve işe girmemde yardımcı oldu. Ayrıca, öğrenci kulüplerindeki çalışanlara da teşekkürler.

Son olarak, danışmanım Doç. Dr. Peyman MAHOUTI'ye teşekkür etmek istiyorum. Öğretmenimiz okula geldiğinden beri, derslerini dinlemekten keyif aldım. Sohbetvari, bilgilendirici öğretme tarzı ve derslerinin modern teknolojilerle ilişkisi, derslerini takip etmeyi daha keyifli hale getirdi. Aslında, mikrodalga ile ilgili negatif düşüncem tamamen değişti ve sonuç olarak anten modellemesi üzerine mezuniyet tezimi yazmak istedim.

Bu mezuniyet tezı, derin öğrenme algoritmalarının yardımıyla helix antenlerin optimizasyonunun tasarımını içermektedir. Şu anda, AI araçlarının ve chatbotların yayınlanmasıyla birlikte teknoloji gelişimi tekrar hızlanmıştır. Yakın gelecekte, antenlerin tasarımı için çok daha verimli yöntemler kullanılabilir. Bu tezin, araştırmacılara heliks anten tasarımı ve optimizasyonu konusunda ek bilgi sağlayabileceğini umuyoruz.

Haziran, 2023

Uygar Tolga KARA

TABLE OF CONTENTS

LIST OF SYMBOLS

dB	Decibel
f	Frequency
G	Giga
Hz	Hertz
M	Mega
w	Wavelength

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
FSH	Feed Stub Height
GEO	Geostationary Earth Orbit
GPR	Ground Plane Radius
HEO	Highly Elliptical Orbit
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
VSWR	Voltage Standing Wave Ratio

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

ARTIFICIAL INTELLIGENCE ASSISTED SURROGATE MODELING AND OPTIMIZATION OF HELIX ANTENNA DESIGN FOR SATELLITE MISSIONS

Uygar Tolga KARA
Department of Aviation Electrics and Electronics
Graduation Thesis

Advisor: Assoc. Prof. Peyman MAHOUTI

Helix antennas are used for different purposes in ground and space applications. In space, they are commonly used in weather, amateur radio, research, communication, and space exploration satellites due to their circular polarization, low latency, and robust design. In all space applications designing, simulating, and optimizing a helix antenna takes time, costs money, requires high computing power and requires usage of modern methods. In this science paper, we aim to reduce time needed for design, simulation, and optimization of a helix antenna with accuracy requirement in mind. Due to high computing time for frequency values over 6 GHz, we aimed to analyze inside a frequency range between 0.3 GHz and 3 GHz, but due to high VSWR values under 0.5 GHz and low power values over 2 GHz, we have changed our frequency range from 0.5 GHz to 2 GHz in steps of 0.1GHz. Also, due to the lower effect of FSH parameter on various output types, we have changed that parameter from varying to constant parameter. Due to the compatibility and low MSE values, we have used ANN MLP deep learning model and gathered MSE of ~0.043 and we have managed to reduce MSE to ~0.037 after optimization with 10 hidden layer size with trainscg - train function. Also, we have performed iteration time analysis with accuracy and simulation time comparison. In the modern age where technology advancement rate has started to increase exponentially again, we hope that this paper may give insights to researchers.

Keywords: MATLAB, antenna, helix, modelling, simulation, optimization, deep learning

YILDIZ TECHNICAL UNIVERSITY

DEPARTMENT OF AVIATION ELECTRICS AND ELECTRONICS

YAPAY ZEKA DESTEKLİ SURROGATE MODELLEME VE UYDU GÖREVLERİ İÇİN HELİS ANTEN TASARIMININ OPTİMİZASYONU

Uygar Tolga KARA

Havacılık Elektrik ve Elektroniği Bölümü

Bitirme Tezi

Danışman: Doç. Dr. Peyman MAHOUTİ

Helis antenler, karada ve uzay uygulamalarında farklı amaçlar için kullanılır. Uzayda, dairesel polarizasyonları, düşük gecikme süreleri ve sağlam tasarımları nedeniyle hava durumu, amatör radyo, araştırma, iletişim ve uzay araştırma uydularında yaygın olarak kullanılırlar. Tüm uzay uygulamalarında bir heliks antenin tasarlanması, simüle edilmesi ve optimizasyonu zaman alır, para mal olur, yüksek hesaplama gücü gerektirir ve modern yöntemlerin kullanılmasını gerektirir. Bu bilim makalesinde, bir heliks antenin tasarım, simülasyon ve optimizasyon sürecinde gereken süreyi, doğruluk gereksinimini göz önünde bulundurarak azaltmayı hedefliyoruz. 6 GHz üzeri frekans değerleri için yüksek hesaplama süresi nedeniyle, 0.3 GHz ile 3 GHz arasında bir frekans aralığında analiz yapmayı hedefledik, ancak 0.5 GHz altında yüksek VSWR değerleri ve 2 GHz üzeri düşük güç değerleri nedeniyle frekans aralığımızı 0.5 GHz'den 2 GHz'ye, 0.1GHz adımlarla değiştirdik. Ayrıca, FSH parametresinin çeşitli çıktı türleri üzerindeki düşük etkisi nedeniyle, bu parametreyi değişken olan parametreden sabit olan parametreye çevirdik. Uyumluluk ve düşük MSE değerleri nedeniyle, ANN MLP derin öğrenme modelini kullandık ve ~0.043 olan MSE'yi, 10 gizli katman boyutu ve trainscg - eğitim fonksiyonu ile optimizasyon sonrası ~0.037'ye indirmeyi başardık. Ayrıca, doğruluk ve simülasyon süresi karşılaştırmasıyla iterasyon süresi analizi gerçekleştirdik. Bu makalenin araştırmacılara içgörü sağlamasını umuyoruz.

Anahtar Kelimeler: MATLAB, anten, helis, modelleme, simülasyon, optimizasyon, derin öğrenme

YILDIZ TEKNİK ÜNİVERSİTESİ

HAVACILIK ELEKTRİK VE ELEKTRONİĞİ BÖLÜMÜ

SECTION 1

INTRODUCTION

1.1 Context and Overview

In history, satellites are used for many tasks. One of the tasks is communication. Signals needs to be sent and received. But signals lose strength when passing through objects or goes for long distances. Earth is not flat and objects are present between antennas. Hence satellites are used with the case that there are no objects between satellites and ground stations when they are on visible viewpoint together. Satellites needs to contain antennas for receiving and sending signals and antennas may be different type based on purpose of the satellite.

As can be seen in Table 1.1, there are many different usage of satellites. In this paper, we are focusing on use of helical antennas which have advantages for usage in weather satellites, communication satellites, scientific research satellites and space exploration satellites. Some advantages of helical antennas are:

- Circular polarization
- Wide frequency band operation capability
- Robustness and simple construction
- Capability for highly directional radiation pattern
- Flexible and customizable design

We may only have one opportunity to launch a satellite. Hence, we need to be sure that the antenna we are attaching works without problem and gives required performance. For this purpose we need to design, simulate and optimize helix antenna. In this paper, we are focusing on design, simulation and optimization of a helix antenna with focus on reducing cost and getting better performance with requirements.

Table 1-1 Satellite Types with Helix Antenna

Satellite Type	Altitude	Antenna Type	Frequency Range
Weather	LEO	Helical, Parabolic	1.7 GHz, 8 GHz
Communication	GEO	Helical, Parabolic, Horn, Phased Array	4-8 GHz, 12-18 GHz, 26.5-40 GHz
Scientific Research	LEO, MEO, HEO	Helical, Dipole, Horn Reflector	300 MHz - 3 GHz, 26.5 - 40 GHz
Space Exploration	Beyond Earth	Helical, High gain antennas	8 - 12 GHz, 26.5 - 40 GHz

From these purposes space exploration satellite UHF band usage gave us inspiration and helped us to complete the project in time due to smaller computing power and runtime compared to other frequency bands. Also we are on 1.7 GHz frequency as well.

1.2 Objective of Study

In this paper, we are aiming to complete tasks such as:

- Perform iteration time and parameter analysis
- Prepare a deep learning AI model with MSE lower than 5
- Optimize deep learning model with lower MSE as result
- Compare runtime of different processes

1.3 Problem Definition

Before launching a satellite containing a helix antenna, we need to design an antenna which satisfies requirements and gives as good performance without going over cost limit. To construct the antenna, we need to follow these steps:

- Analysis of mission and stakeholder requirements

- Design and modelling of antenna
- Antenna simulation and optimization
- Producing first prototype
- Testing prototype on laboratory
- Revision in Case of Problems
- Performing environmental and reliability tests
- Performing integration and system tests
- Launching satellite and orbit verification

In this paper, we are focusing on first 3 steps. Total required time for these steps may take nearly 5 to 10 weeks. In this time range monetary costs related to personnel wage, software, facility and equipment may occur. As a estimate this process may take 130,000 to 200,000 dollars. So, design steps do require time and money. We also need to keep in mind that this estimation is for a process without any problem. With these things in mind, our aim is to use AI algorithms and methods to reduce time and cost in design, simulation and optimization phase of helix antenna construction.

1.4 Problem Importance

Helix antennas are commonly used in weather satellites, communication satellites, scientific research satellites and space exploration satellites. All of these purposes are important for daily information or researching new technologies. Also, from data, we know that there are nearly 7702 satellies on orbit. Hence we need to complete satellite on a limited time. In satellite, designing, simulating and optimizing an antenna is a time consuming and costly process. If we encounter problems, time need and cost may increase and it may be harder to finish project on time. Hence, modelling, simulation and optimizing processes needs to be passed without problem in a time window. We are aiming to prepare AI models that can help solving complex problems in short time.

1.5 Software Overview

In this paper, we are aiming to use these software products:

- MATLAB (Antenna Toolbox, Statistics Toolbox, Deep Learning Toolbox)
- Python (Not planned to use, but we may use for missing AI algorithms)
- ANSYS HFSS (Not planned to use, but we may use for more simulation)

As general information, we are not planning to produce any real parts.

SECTION 2

LITERATURE REVIEW

2.1 Literature Review

We have researched previous work done for helix antenna design and in our 10 references, we have learned different viewpoints.

A reference named “Guidelines for Design of Helical Antennas” [1] gives a very good guidance on helix antenna modeling. In this source we may see guidelines for nonuniform helical antenna design with analytical viewpoint. In this paper we are focusing on design of uniform helical antennas. But we can learn compatible information from that source. For example, we may see that maximizing gain in main radiating direction is an important output with circular polarization in mind. We may also see that in that paper, they have worked on operating frequencies from 30 MHz to 6 GHz. It is also conventional to make all optimizations on 300MHz single frequency. In the paper wavelength-gain xy graphs with changing radius with different coefficient with wavelength are presented. Generally larger radius usage gives higher gains but there are certain wavelengths where gain values come closer, hence it may be advantageous tos select lower wavelength on cases. As final note we may see that in their optimized antenna, operating frequency is 3GHz, wire material is copper, and they have achieved a high antenna gain of 16.7 dB.

In next reference [2] we may see that they have worked with microstrip antenna structure and selected operating frequency range as 240-360 MHz. They made frequency-resistance, impedance, s parameters graphs and we may see that most optimum results were gathered on and/or near 300 MHz frequency.

In our next reference [3], in a different type of application, we may see that operating frequeny range is selected between 4.4 GHz to 5.6 GHz and optimum result was gathered on 5 GHz. In another source [4] we may see that directivity helical antenna is linearly increasing with the number of helix turns. In previous sources, we also saw that taking number of turns more than 8 (less than 12) may be an approach for some outputs. With increasing turns, we reduce width of the ptimum helix antenna.

Another research [5] gives very clear steps and viewpoint for simulating an dipole helix antenna in between 440MHz to 480MHz with operating frequency at 450MHz. Best results were gathered at 456 MHz. 15 turns were used.

In another work [6] where helical antenna is designed for underwater application, it is observed that gain increase with number of turns and also height of the antenna increases gain. Wire thickness effect antenna performance in smaller effect. In here as a personal opinion taking 10 as number of turns does look advantageous on S11 output.

In next research [7] we may see that larger frequency range between 0GHz-10GHz is taken from 4.78 GHz to 6.21 GHz antenna has input impedance bandwidth which is giving %26 fractional bandwidth. Axial ratio below 3dB is gathered between 5.23 GHz to 5.64 GHz. From gain plot, 5-6 GHz seems to give larger gains.

In another informative research [8] informs that in optimized antennas, taking turn number higher than 6 gives better figures. Also taking turn number below 7, had small deterioration in antenna gain and radiated power with antenna length reduced. It also recommends that if circular polarization is a key requirement, specifying axial ratio needs to be set as constraint. As general personal opinion taking n as 7 looks advantageous in directed power.

In last second reference [9] gives an simulation time graph for different phases. Our antenna is different but surrogate modeling method and time table is taken as a reference to add in out research paper. In this paper optimization phase took 120 hours. In our paper we are going to try to give simulation time for optimization.

Last reference [10] is our main reference. Our research paper inspired from this reference directly. As similar, we are aiming for Data Driven Surrogate Assisted Optimization of Helix Antenna Using Deep Learning. Neural networks does help reduce optimization time effectively. Generally 2-10 GHz is taken as frequency range. In this paper, using deep learning based model M2LP using HBMO algorithm reduced computational cost of the optimization procedure is reduced by almost 90%. Results are shown in frequency graphs and accuracies are compared for different models.

SECTION 3

DATA DRIVEN SURROGATE MODEL OF HELIX ANTENNA

3.1 Helix Antenna Parameters

A helix antenna may be generally analyzed with these parameters and outputs:

- Turn number: Number of wire turns on the antenna (clockwise or reverse)
- Radius: Value of radius of turns. (antenna may be uniform or non uniform)
- Width: Wire strip width value. Has a high effect on directivity and S11 mag.
- Spacing: Value of vertical space between wire turns.
- Feed Stub Height: Distance from ground to feeding stub.
- Ground Plane Radius: Radius of ground plane that antenna is on.
- Material: Type of material of helix antenna. (Copper, PEC...)
- Substrate: Type of dielectric material of antenna. (Air, water...)
- Pitch Angle: Angle between tangent and parallel plane.
- Axial Ratio: EF major axis to EF minor axis ratio. (Towards 1)
- Operating Frequency: Efficient operating frequency for antenna
- Input Impedance: Impedance of antenna’s input terminals
- Radiation Pattern: Distribution of radiated power around antenna
- Gain and Realized Gain: Power increase in peak radiation direction

Note that there are many more parameters and outputs from a helix antenna.

3.2 Helix Antenna AI Algorithms

These are possible AI methods to use for design of a helix antenna:

- Genetic Algorithms: Mimic natural selection. May be used for optimization.
- Particle Swarm Optimization: It is a population based optimization algorithm.
- Artificial Neural Networks: Used in modeling and optimization of antenna.
- Support Vector Machines: Supervised algorithm for predicting performance.
- Deep Learning: Uses ANN and big data for design and performance prediction.

In this project, we are using a deep learning method (ANN MLP) for antenna modeling and performance prediction. This method worked compatible with current problem, but in another case we may need to use another method. Also, we would like to mention that using neural networks reduce computing time by high percentage compared to manual iteration and may give a good accuracy value with help of data.

3.3 Simulation Iteration Time Analysis

Before starting this subsection, we would like to mention that all MATLAB codes for all subsections may be found in Appendix A. Before making an AI model, we would like to make 2 analysis actions. Firstly, we measured the iteration time for manual simulation of a helix antenna while the frequency values change between a range with fixed step value. As result, we have gathered the following iteration times as minutes:

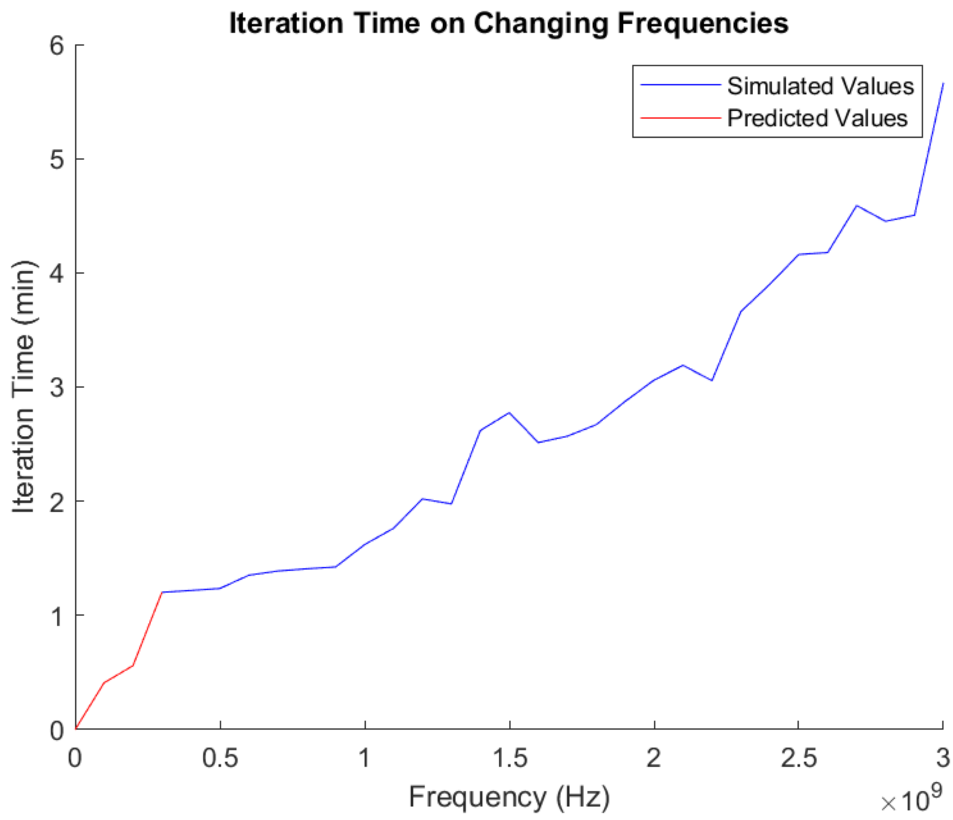

Figure 3-1 Iteration Time in Frequency Range

From figure 3-1, we may see that minimum iteration time on lowest frequency value (0.3GHz) is over 1 minutes and highest iteration time on highest frequency value (3GHz) is over 5 minutes. As a general relationship, we may see that iteration time increase with the increase of frequency so that we need more computing time.

In this analysis we took frequency range from 0.3GHz to 3GHz with steps of 0.1GHz. But we did not know iteration times for frequencies below 0.3GHz. For this situation, we assumed that the iteration time for frequencies close to zero is also close to zero. After this assumption, we made a regression model and used the model.

From this graph we may make an estimation for a manual simulation (for loop) of helix antenna. Assume we change 6 parameters (individually specify 5 values for each parameter) and we also change frequency as mentioned. (28 values) Hence in nested for loops, iteration number is: $5^6 * 28 = 437500$ combinations. If we assume average iteration time is 2.5 minutes, total computing time is nearly 760 days. It is nearly 2 years. Hence, we need to reduce computing time by using deep learning algorithms.

3.4 Parameter Boundary Values and Parametric Analysis

Before sharing our new parametric analysis results, we would like to share our findings from our previous work on different frequency values and an output type. In our past and current project, we have used 6 parameters as varying parameters. Lower and upper limits with constant values of these parameters are shared in following areas.

Table 3-1 Parameter Coefficient Effect Analysis

Parameter Name	Effect on S11 Magnitude
Turns	0.08
Radius	5.21
Width	82.79
Spacing	9.60
FSH	0.07
GPR	2.26

In Table 3-1, 6 parameters are varying parameters we use in our project as well. In the table 3-2, we may see width has maximum effect on S11 magnitude in parameters.

Table 3-2 Boundary and Constant Values of Varying Parameters

Parameter	Lower Bound	Upper Bound	Constant	Main Effects
Turns	3	15	9	axial ratio, gain, and bandwidth
Radius	w/5	w/3	w/4	input impedance and radiation pattern
Width	w/100	w/50	w/75	impedance matching and bandwidth
Spacing	w/4	w/2	w/3	axial ratio and bandwidth
FSH	w/1000	w/100	w/550	radiation pattern
GPR	w/2	w*2	W*1.25	radiation pattern and input impedance

Currently, we picked typical values or average / balance values for middle effect.

In our current project, we performed parametric analysis on these output types:

- Maximum gain
- Maximum realized gain
- S11 Magnitude (dB)
- VSWR
- Return loss
- Power (dB)
- Electric field max. Magnitude

For method, we only changed 1 out of 6 parameters and took 5 parameters as constant. For remaining parameter, we made multiple line graphs while changing frequency. This helped us see the effect of parameter on output visually.

Graphs for each parameter analysis may be found in following subsections. We would like to write our thoughts and decisions about parametric analysis cases.

- For turns parametric analysis, it was our most detailed analysis. Specifically for turns wise we did not spot an unusual case. We think taking turn number as 7 gave advantage on S11 magnitude and return loss. Parameter effect is medium.
- For radius parametric analysis, taking radius as 0.06 gave some more advantages on gain. No other unusual effect are seen. Effect is medium/high.
- For width parametric analysis, taking width higher gave advantage on some outputs. No other unusual effect are seen. Parameter effect is low/medium.
- For spacing parametric analysis, higher values gave unstable results on higher frequencies. Hence lower value is advised. Parameter effect is medium/ high.
- For FSH, with the effect of our narrow parameter range selection, we can see that the effect is low. Hence we have decided to take FSH as constant.
- For GPR parametric analysis, no unusual effect are seen. Effect is low/medium.

Beside individual parametric analysis graphs, we also saw recurring situations in general. With these recurring effects, we decided to take following actions in future:

- We changed lower frequency bound from 0.3GHz to 0.5GHz, because lower values gave very high VSWR values. As mentioned we removed FSH too.
- We changed higher frequency bound from 3GHz to 2GHz, because we repetitively saw lower values after 2GHz on power and Efield magnitude.

3.4.1 Turns Parametric Analysis

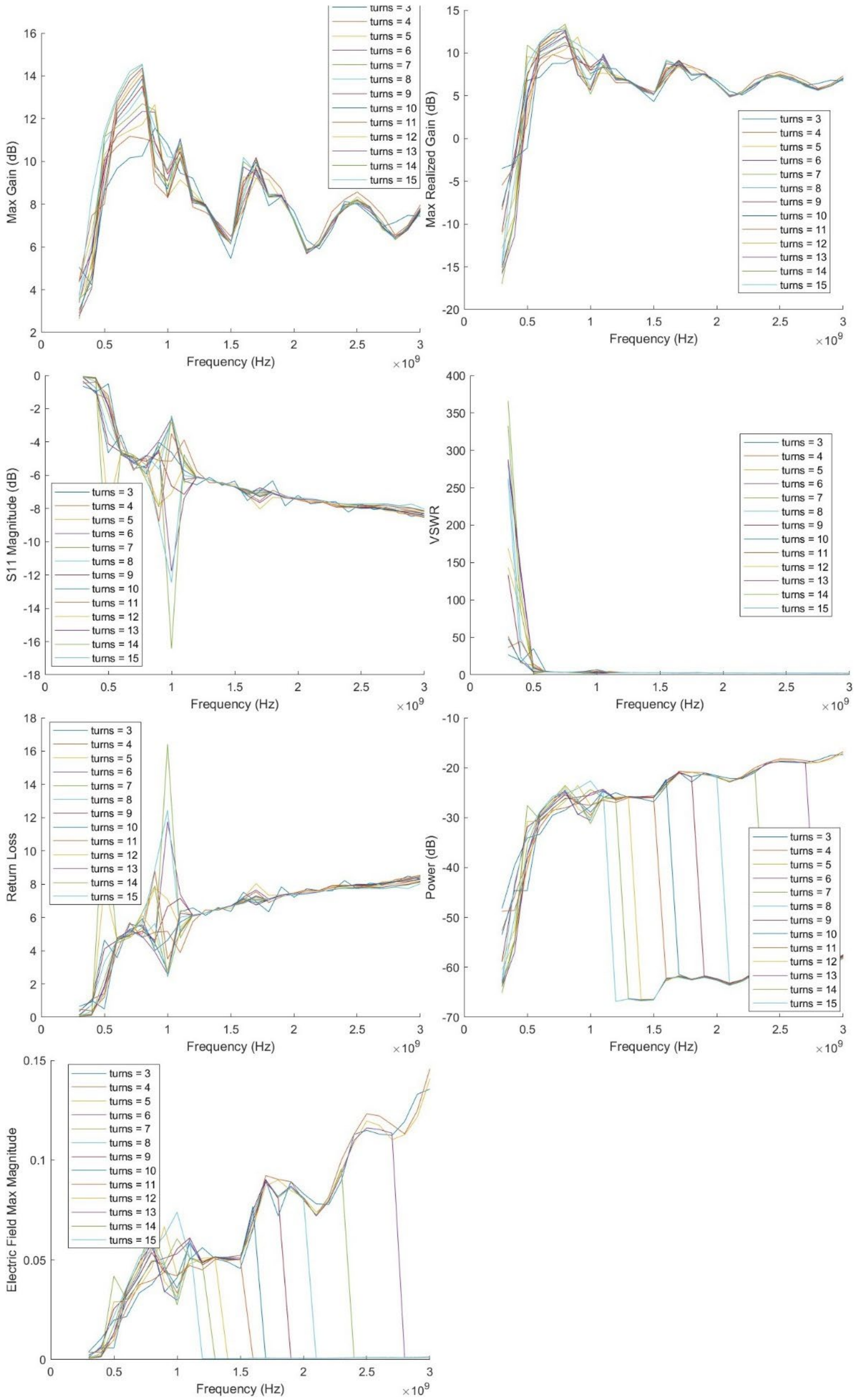

Figure 3-2 Parametric Analysis of Turns

3.4.2 Radius Parametric Analysis

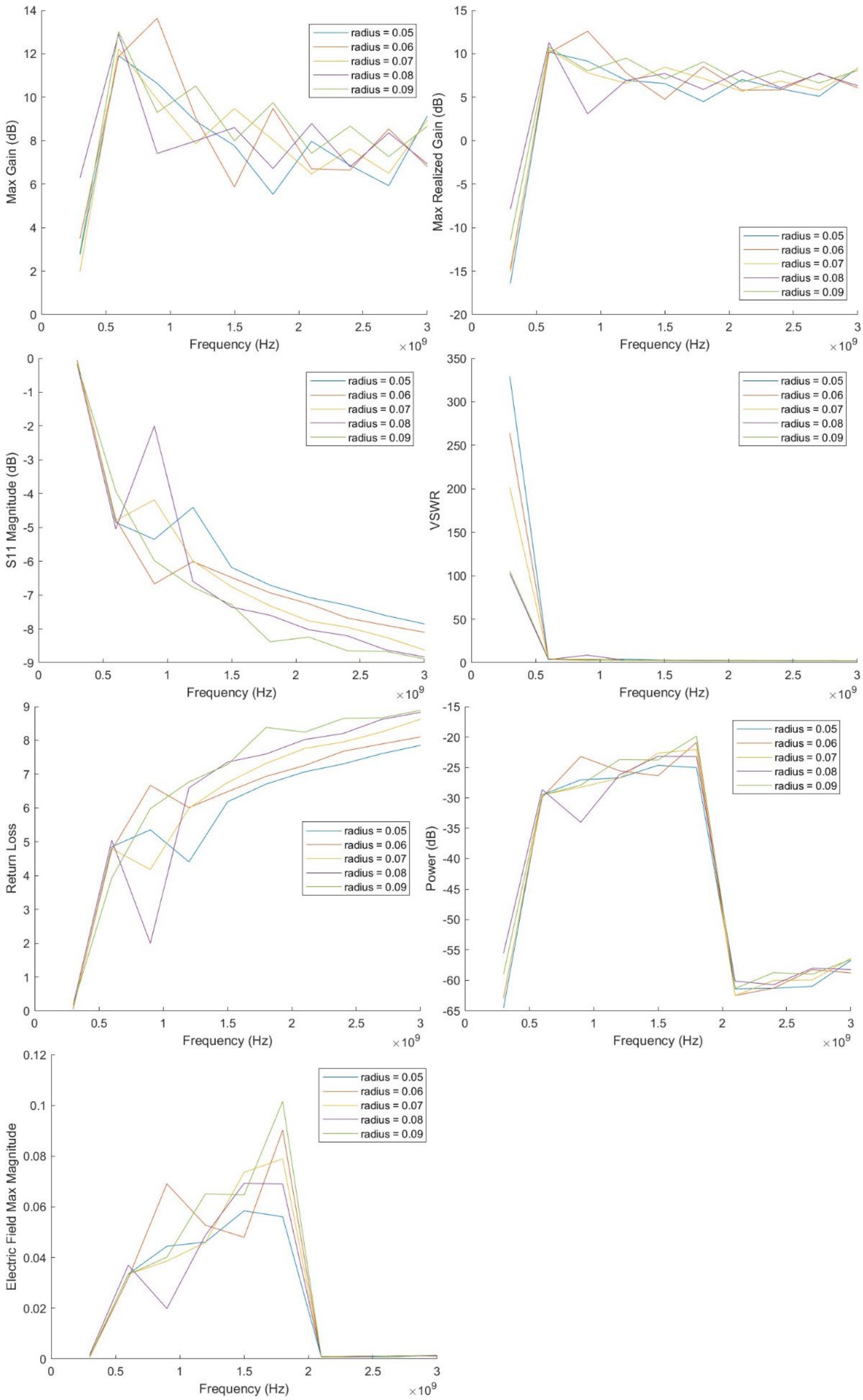

Figure 3-3 Parametric Analysis of Radius

3.4.3 Width Parametric Analysis

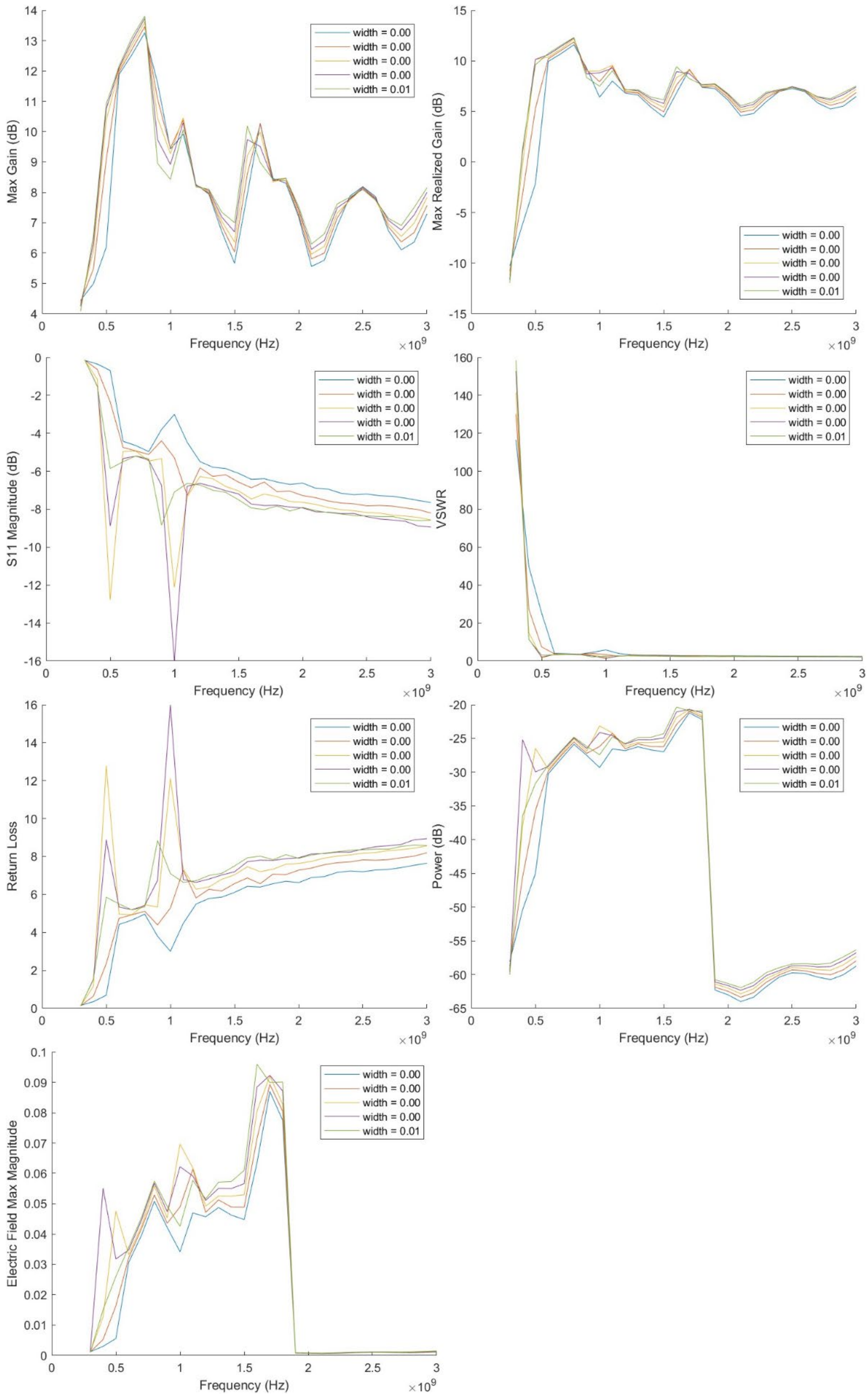

Figure 3-4 Parametric Analysis of Width

3.4.4 Spacing Parametric Analysis

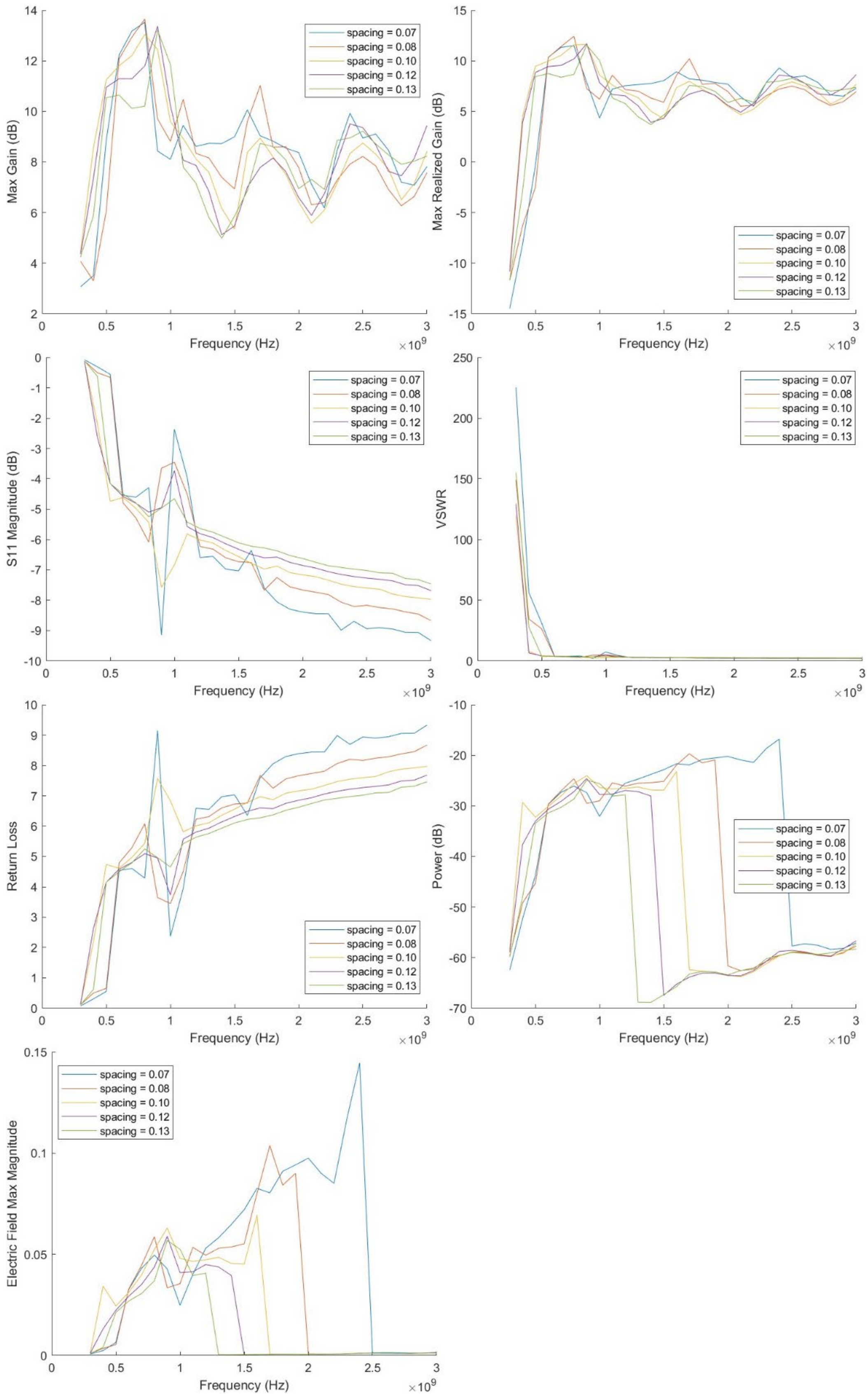

Figure 3-5 Parametric Analysis of Spacing

3.4.5 FSH Parametric Analysis

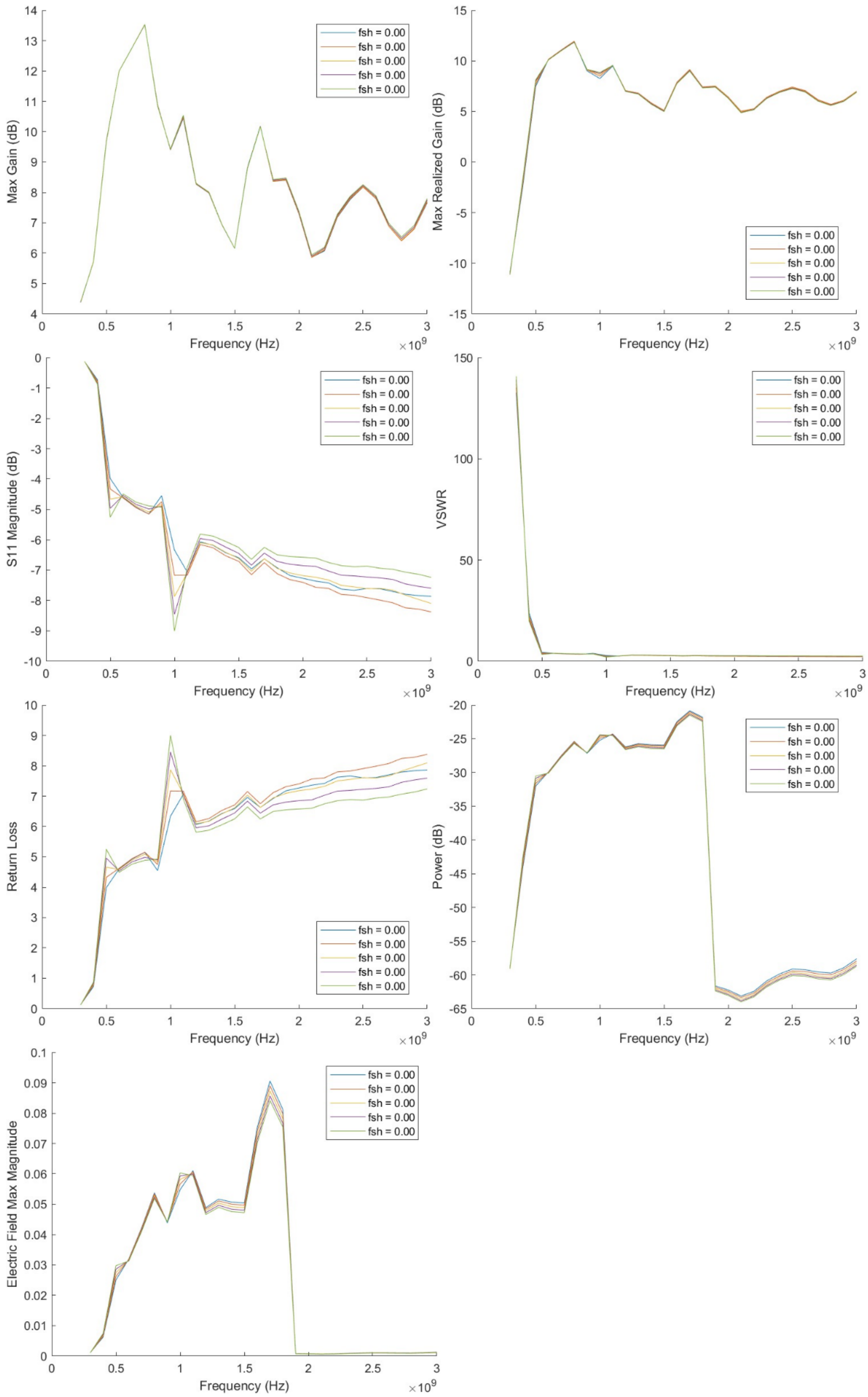

Figure 3-6 Parametric Analysis of FSH

3.4.6 GPR Parametric Analysis

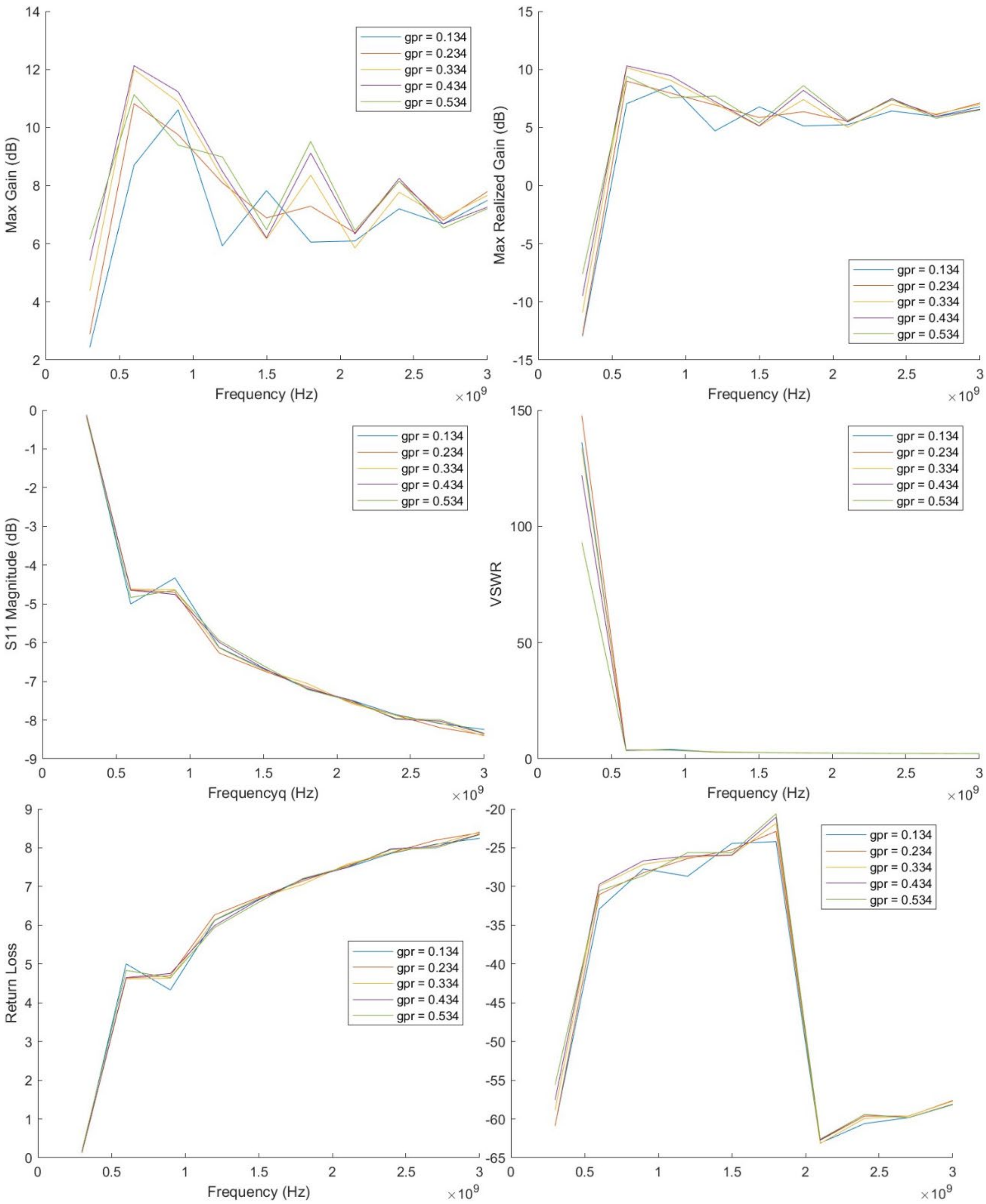

Figure 3-7 Parametric Analysis of GPR

Figure 3.4.1, 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.6 are analyzed before the figures and in section 3.4. Figures are added for visualizing the parameter effects on outputs.

3.5 Building Deep Learning AI Model

After parametric analysis, we made changes for our parameters as mentioned. Now, we need to generate a random dataset and using training and test data values, we need to prepare an AI model using ANN MLP algorithm and measure accuracy. (MSE)

As mentioned, all MATLAB codes can be found in Appendix A. AI model attributes are:

- We used 2 hidden layers with 10 neurons in each layer.
- We used logsig as the activation function.
- We have prepared 100 sample dataset and used %80 as training data.

Reason for using sample number as 100 was to finish AI model in time. We would recommend taking sample number as 1000 in future. Dataset generation took more than 202 minutes to complete. AI model generation took several seconds. We did get MSE value as 0.043469 hence we did not increase layer and neuron number.

3.6 Optimizing Deep Learning AI Model

We started optimization in aim of getting lower MSE value. Properties are as:

- With optimization MSE value has been reduced from 0.043 to 0.036.
- Process took more than 1 minutes and 30 seconds.
- Optimal hidden layer size has been found as 10.
- Optimal training function has been found as trainscg.
- Process stopped at 30 iterations, 30 epochs.

With this step we have generated 2 network models. (Default AI model and optimized AI model) These models can later be used for performance prediction, parameter generation for a scenario, optimization and for other purposes.

With this step, we have completed modelling, simulation and optimization of a helix antenna. We would like to share graphs as details about optimization process. In figure 3-8 fit graph may be seen and in figure 3-9 progressive objective function is given.

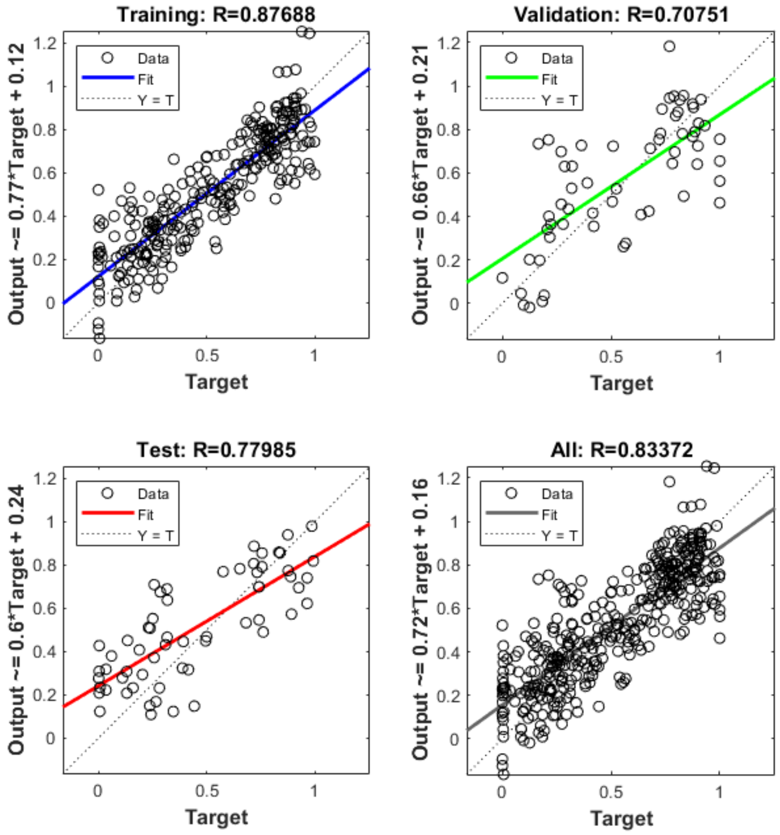

Figure 3-8 Fit Line Graphs of Optimization Process

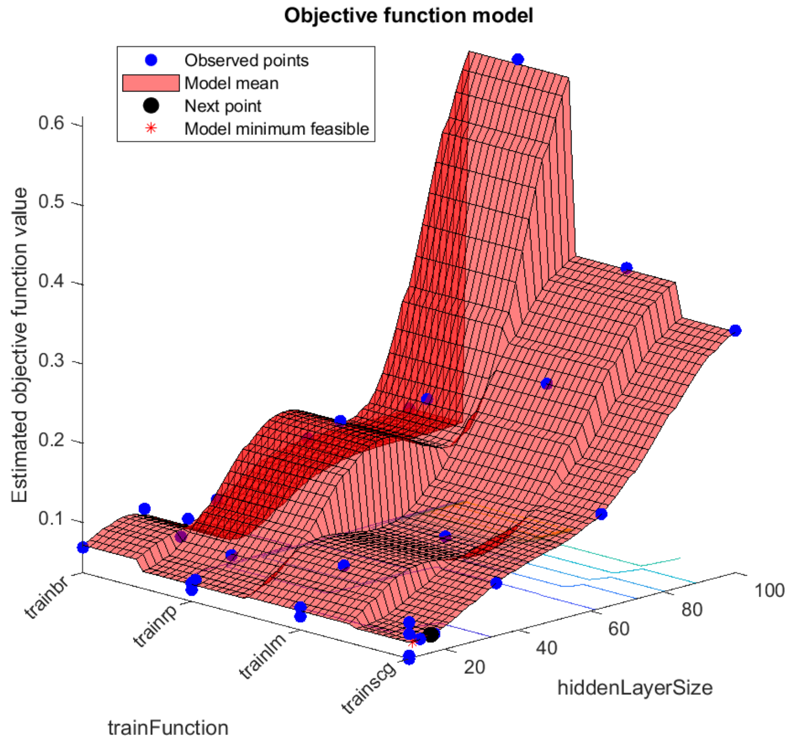

Figure 3-9 Objective Function Model at Optimization

3.7 Extra Section – Parameter Generation Use Case for AI Model

In this extra section, we would like to share our previous work about parameter generation for requirement of max gain and max S11 magnitude is requested.

In this scenario we have another ANN MLP model which have higher MSE:

- Maximum gain and S11 magnitude requested at same time
- Operating frequency is given as 300 MHz, hence 0.3GHz
- ANN MLP model with Global Optimization Toolbox is used.

Table 3-3 Parameter Selection Scenario Optimization

Generation	Func-count	Best f(x)	Mean f(x)	Stall Gen.
1	400	-9.498	6.713	0
2	590	-9.498	1.295	1
3	780	-9.498	-0.7207	2
4	970	-9.548	-3.583	0
5	1160	-9.621	-4.444	0
6	1350	-9.649	-6.304	0
7	1540	-9.705	-7.56	0
8	1730	-9.707	-8.599	0
9	1920	-9.707	-9.206	1
10	2110	-9.818	-7.964	0

After completion of process in table 3-3, we have gathered parameters as:

- Turns 11
- Radius 0.1289
- Width 0.0648 (changed to 0.0148 because of parameter relationships)
- Spacing 0.0516
- FSH 0.025
- GPR 0.0078 (changed to 0.1578 because of parameter relationships)

We did not get a stable antenna but some results are showed in figure 3-10, 3-11.:

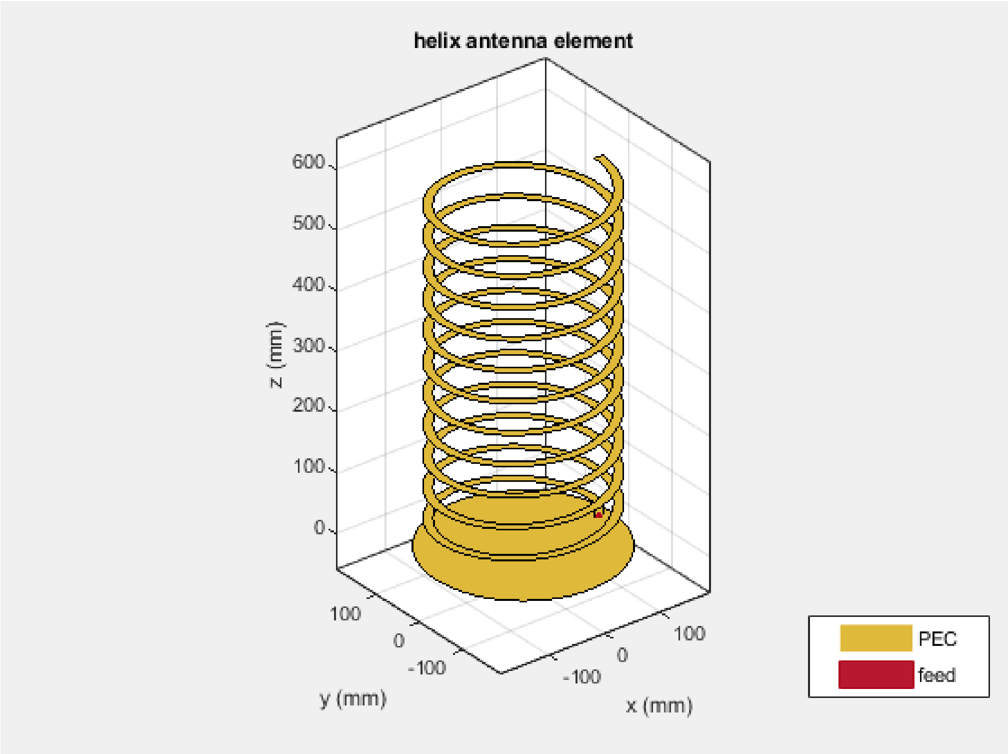

Figure 3-10 Automatic Parameter Selected Helix Antenna

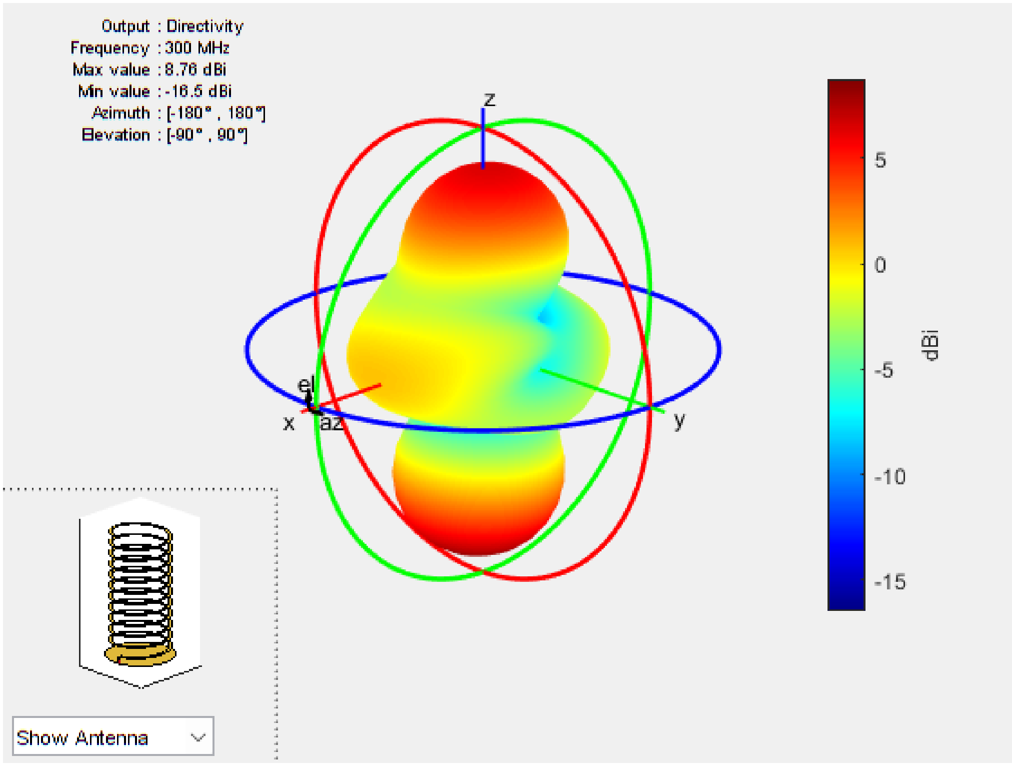

Figure 3-11 Radiation Pattern of Helix Antenna

RESULTS AND DISCUSSIONS

In this section we would like to share computing time results for processes in table 4-1:

Table 4-1 Processes Computing Time Comparison

Process Name	Process Detail	Computation Time
Iteration Time Analysis	28 simulation iterations	78.01 minutes
Parametric Analysis	684 simulation iterations	42.21 hours
Generate Dataset	100 simulation iterations	202.35 minutes
ANN MLP Model Build	2 hidden layers, 100 sample number, %80 training, %20 test Training function logsig MSE as 0.04346	10 seconds
ANN MLP Optimization	10 hidden layers Training function trainscg 30 epochs, 30 iterations MSE as 0.03677	90 seconds
Extra Case, Automatic Parameter Selection	Maximum S11 magnitude and maximum gain	54 minutes
Nested for loop simulation	437500 iterations	760 days

SECTION 5

CONCLUSION

We have completed modeling, simulation and optimization of a helix antenna with help of ANN MLP deep learning algorithm and MATLAB. In conclusion, we may see that using deep learning algorithms instead of using nested for loop simulation, reduces computing time for more than %90. Reducing computing time also reduces project cost as well. In this current era where technology advances exponentially again, we hope that this paper may give insights for people who want to model helix antennas for purposes.

REFERENCES

[1] J. Dinkic, D. Olcan, A. Djordjevic, "Guidelines for design of helical antennas," in 2022 30th Telecommunications Forum (TELFOR), 2022, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/TELFOR56187.2022.9983713

[2] S. L. S. Kilambi, M. P. Da Cunha, "Design and measurements of a small normal mode helical antenna with integrated microstrip structure," in 2022 Antenna Measurement Techniques Association Symposium (AMTA), 2022, pp. 1–6. [Online]. Available: http://dx.doi.org/10.23919/AMTA55213.2022.9955019

[3] R. K. Singh, N. P. Pathak, "Design and performance evaluation of axial mode helical antenna in cylindrical water pipe network having irregular internal depositions," in 2021 IEEE Indian Conference on Antennas and Propagation (InCAP), 2021, pp. 117–120.[Online].Available: http://dx.doi.org/10.1109/InCAP52216.2021.9726361

[4] O. A. Ella, "New design rules to improve helical antenna performance," in 2021 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), 2021, pp. 248–252. [Online]. Available: http://dx.doi.org/10.1109/MTTW53539.2021.9607134

[5] A. D. M. Africa, J. M. C. Lacanilao, R. V. A. B. Lamdagan, "Design of a helical antenna for handheld transceivers," in 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), 2020, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/HNICEM51456.2020.9400052

[6] A. Jaafar, H. Ja'afar, Y. Yamada, F. Sadeghikia, I. Ibrahim, M. Mahmood, "Design of an axial mode helical antenna with buffer layer for underwater applications.," International Journal of Electrical and Computer Engineering, vol. 13, no. 1, pp. 473-482, 2023. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85143861272&lang=tr&site=eds-live&scope=site&authtype=ip,uid

[7] A. K. Pandey, S. K. Pathak, "Design of normal mode circularly polarized helical antenna at 5.3 ghz," in 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), 2020, pp. 1–4. [Online]. Available: http://dx.doi.org/10.1109/UPCON50219.2020.9376491

[8] C. L. Sabioni, W. Tiano Dos Santos Ramos, J. A. de Vasconcelos, E. J. da Silva, R. Adriano, R. R. Saldanha, "Multiobjective design optimization of non-uniform

helical antennas," in 2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), 2015, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/IMOC.2015.7369169

[9] S. Koziel, M. A. Belen, A. Çali¸skan, P. Mahouti, "Rapid design of 3d reflectarray antennas by inverse surrogate modeling and regularization," IEEE Access, vol. 11, pp. 24 175–24 184, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3254204

[10] P. Mahouti, A. Belen, O. Tari, M. A. Belen, S. Karahan, S. Koziel, "Data-driven surrogate-assisted optimization of metamaterial-based filtenna using deep learning," Electronics, vol. 12, no. 7, p. 1584, Mar. 2023. [Online]. Available: http://dx.doi.org/10.3390/electronics12071584

[11] NanoAvionics. (2023, May 4). How Many Satellites are in Space? Retrieved June 12, 2023, from https://nanoavionics.com/blog/how-many-satellites-are-in-space/

APPENDIX-A

MATLAB CODE

```
%% Parameters

clear; clc;
frequency =  0.3e9 : 0.1e9 : 3e9;
wavelength = freq2wavelen(frequency);

turns_constant = 9 ;
radius_constant = mean(wavelength) / 4;
width_constant = mean(wavelength) / 75;
spacing_constant = mean(wavelength) / 3;
fsh_constant = mean(wavelength) / 550; % feedstubheight
gpr_constant = 1.25*mean(wavelength); % groundplaneradius

%% Iteration Time Analysis

storage1 = zeros(size(frequency));

for j = 1:length(frequency)

    tic;

    f = frequency(j);

    hx = helix("Turns", turns_constant, "Radius", radius_constant, "Width",
width_constant, ...
    "Spacing", spacing_constant, "FeedStubHeight", fsh_constant,
"GroundPlaneRadius", gpr_constant);

    max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
    max(max(pattern(hx,f,-180:180, -90:90,"Type","realizedgain")));
    20*log10(abs(sparameters(hx,f).Parameters)); % s11 magnitude (dB)
    (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
    -20*log10(abs(sparameters(hx,f).Parameters)); % return loss
    max(max(pattern(hx,f,-180:180, -90:90,"Type","powerdb")));
    max(max(pattern(hx,f,-180:180, -90:90,"Type","efield")));

    storage1(j) = toc;

    fprintf("Frequency %d/%d \n", j, length(frequency));

end

plot(frequency, storage1); xlabel('Frequency (Hz)'); ylabel('Iteration Time
(sec)');

% adding and plotting missing frequencies
storage1_minutes = [0, storage1_minutes];
frequency_temp = [0, frequency];
p = polyfit(frequency_temp, storage1_minutes, 1);
missing_frequencies = [0.1e9, 0.2e9];
predicted_minutes = polyval(p, missing_frequencies);
```

```
frequency_temp = [frequency_temp(1), missing_frequencies,
frequency_temp(2:end)];
storage1_minutes = [storage1_minutes(1), predicted_minutes,
storage1_minutes(2:end)];

figure;
hold on;
plot(frequency_temp(4:end), storage1_minutes(4:end), 'b');
plot(frequency_temp(1:4), storage1_minutes(1:4), 'r');
xlabel('Frequency (Hz)');
ylabel('Iteration Time (min)');
title('Iteration Time on Changing Frequencies');
legend('Simulated Values', 'Predicted Values');
hold off;

%% Turns parametric analysis

tic; n=13;

turns = round(linspace(3,15,n));
legend_turns = cell(size(turns));

figure(1); hold on;
figure(2); hold on;
figure(3); hold on;
figure(4); hold on;
figure(5); hold on;
figure(6); hold on;
figure(7); hold on;

for i = 1:length(turns)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        t = turns(i); f = frequency(j);

        hx = helix("Turns", t, "Radius", radius_constant, "Width",
width_constant, ...
        "Spacing", spacing_constant, "FeedStubHeight", fsh_constant,
"GroundPlaneRadius", gpr_constant);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return
loss
```

```
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(turns), j,
length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_turns{i} = sprintf('turns =
%d', t);
    figure(2); plot(frequency, storage2); legend_turns{i} = sprintf('turns =
%d', t);
    figure(3); plot(frequency, storage3); legend_turns{i} = sprintf('turns =
%d', t);
    figure(4); plot(frequency, storage4); legend_turns{i} = sprintf('turns =
%d', t);
    figure(5); plot(frequency, storage5); legend_turns{i} = sprintf('turns =
%d', t);
    figure(6); plot(frequency, storage6); legend_turns{i} = sprintf('turns =
%d', t);
    figure(7); plot(frequency, storage7); legend_turns{i} = sprintf('turns =
%d', t);

end

% hold off;

figure(1); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Gain (dB)');
figure(2); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('VSWR');
figure(5); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Return Loss');
figure(6); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Power (dB)');
figure(7); legend(legend_turns, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% Radius parametric analysis

tic; n=5;

frequency =  0.3e9 : 0.3e9 : 3e9;
radius = linspace(mean(wavelength)/5, mean(wavelength)/3, n);
legend_radius = cell(size(radius));

figure(1); hold on;
figure(2); hold on;
figure(3); hold on;
figure(4); hold on;
figure(5); hold on;
```

```
figure(6); hold on;
figure(7); hold on;

for i = 1:length(radius)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        r = radius(i); f = frequency(j);

        hx = helix("Turns", turns_constant, "Radius", r, "Width",
width_constant, ...
        "Spacing", spacing_constant, "FeedStubHeight", fsh_constant,
"GroundPlaneRadius", gpr_constant);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return
loss
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(radius), j,
length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(2); plot(frequency, storage2); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(3); plot(frequency, storage3); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(4); plot(frequency, storage4); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(5); plot(frequency, storage5); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(6); plot(frequency, storage6); legend_radius{i} = sprintf('radius
= %.2f', r);
    figure(7); plot(frequency, storage7); legend_radius{i} = sprintf('radius
= %.2f', r);

end

% hold off;
```

```
figure(1); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Gain (dB)');
figure(2); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('VSWR');
figure(5); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Return Loss');
figure(6); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Power (dB)');
figure(7); legend(legend_radius, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% Width parametric analysis

tic; n=5;

width = linspace(mean(wavelength)/100, mean(wavelength)/50, n);
legend_width = cell(size(width));

figure(1); hold on;
figure(2); hold on;
figure(3); hold on;
figure(4); hold on;
figure(5); hold on;
figure(6); hold on;
figure(7); hold on;

for i = 1:length(width)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        w = width(i); f = frequency(j);

        hx = helix("Turns", turns_constant, "Radius", radius_constant,
"Width", w, ...
        "Spacing", spacing_constant, "FeedStubHeight", fsh_constant,
"GroundPlaneRadius", gpr_constant);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
```

```
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return
loss
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(width), j,
length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_width{i} = sprintf('width =
%.2f', w);
    figure(2); plot(frequency, storage2); legend_width{i} = sprintf('width =
%.2f', w);
    figure(3); plot(frequency, storage3); legend_width{i} = sprintf('width =
%.2f', w);
    figure(4); plot(frequency, storage4); legend_width{i} = sprintf('width =
%.2f', w);
    figure(5); plot(frequency, storage5); legend_width{i} = sprintf('width =
%.2f', w);
    figure(6); plot(frequency, storage6); legend_width{i} = sprintf('width =
%.2f', w);
    figure(7); plot(frequency, storage7); legend_width{i} = sprintf('width =
%.2f', w);

end

hold off;

figure(1); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Gain (dB)');
figure(2); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('VSWR');
figure(5); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Return Loss');
figure(6); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Power (dB)');
figure(7); legend(legend_width, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% Spacing parametric analysis

tic; n=5;

spacing = linspace(mean(wavelength)/4, mean(wavelength)/2, n);
legend_spacing = cell(size(spacing));

figure(1); hold on;
figure(2); hold on;
figure(3); hold on;
figure(4); hold on;
```

```
figure(5); hold on;
figure(6); hold on;
figure(7); hold on;

for i = 1:length(spacing)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        sc = spacing(i); f = frequency(j);

        hx = helix("Turns", turns_constant, "Radius", radius_constant,
"Width", width_constant, ...
        "Spacing", sc, "FeedStubHeight", fsh_constant, "GroundPlaneRadius",
gpr_constant);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return
loss
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(spacing),
j, length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(2); plot(frequency, storage2); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(3); plot(frequency, storage3); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(4); plot(frequency, storage4); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(5); plot(frequency, storage5); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(6); plot(frequency, storage6); legend_spacing{i} =
sprintf('spacing = %.2f', sc);
    figure(7); plot(frequency, storage7); legend_spacing{i} =
sprintf('spacing = %.2f', sc);

end
```

```
hold off;

figure(1); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Gain (dB)');
figure(2); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('VSWR');
figure(5); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Return Loss');
figure(6); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Power (dB)');
figure(7); legend(legend_spacing, 'Location', 'best'); xlabel('Frequency
(Hz)'); ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% Feedstubheight parametric analysis

tic; n=5;
frequency =  0.3e9 : 0.3e9 : 3e9;

fsheight = linspace(mean(wavelength)/1000, mean(wavelength)/100, n);
legend_fsh = cell(size(fsheight));

figure(1); hold on;
figure(2); hold on;
figure(3); hold on;
figure(4); hold on;
figure(5); hold on;
figure(6); hold on;
figure(7); hold on;

for i = 1:length(fsheight)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        fsh = fsheight(i); f = frequency(j);

        hx = helix("Turns", turns_constant, "Radius", radius_constant,
"Width", width_constant, ...
        "Spacing", spacing_constant, "FeedStubHeight", fsh,
"GroundPlaneRadius", gpr_constant);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
```

```
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 -
abs(sparameters(hx,f).Parameters)); % vswr
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return
loss
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(fsh), j,
length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(2); plot(frequency, storage2); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(3); plot(frequency, storage3); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(4); plot(frequency, storage4); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(5); plot(frequency, storage5); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(6); plot(frequency, storage6); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);
    figure(7); plot(frequency, storage7); legend_fsh{i} = sprintf('fsh =
%.2f', fsh);

end

hold off;

figure(1); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Max Gain (dB)');
figure(2); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('VSWR');
figure(5); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Return Loss');
figure(6); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Power (dB)');
figure(7); legend(legend_fsh, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% GroundPlaneRadius parametric analysis

tic; n=5; frequency =  0.3e9 : 0.3e9 : 3e9;

gpradius = linspace(mean(wavelength)/2, mean(wavelength)*2, n);
legend_gpr = cell(size(gpradius));

figure(1); hold on;
figure(2); hold on;
```

```
figure(3); hold on;
figure(4); hold on;
figure(5); hold on;
figure(6); hold on;
figure(7); hold on;

for i = 1:length(gpradius)

    storage1 = zeros(size(frequency));
    storage2 = zeros(size(frequency));
    storage3 = zeros(size(frequency));
    storage4 = zeros(size(frequency));
    storage5 = zeros(size(frequency));
    storage6 = zeros(size(frequency));
    storage7 = zeros(size(frequency));

    for j = 1:length(frequency)

        gpr = gpradius(i); f = frequency(j);

        hx = helix("Turns", turns_constant, "Radius", radius_constant, 
"Width", width_constant, ...
        "Spacing", spacing_constant, "FeedStubHeight", fsh_constant, 
"GroundPlaneRadius", gpr);

        storage1(j) = max(max(pattern(hx,f,-180:180, -90:90,"Type","gain")));
        storage2(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
        storage3(j) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11 
magnitude (dB)
        storage4(j) = (1 + abs(sparameters(hx,f).Parameters)) / (1 - 
abs(sparameters(hx,f).Parameters)); % vswr
        storage5(j) = -20*log10(abs(sparameters(hx,f).Parameters)); % return 
loss
        storage6(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","powerdb")));
        storage7(j) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","efield")));

        fprintf("Parameter %d/%d - Frequency %d/%d \n", i, length(gpr), j, 
length(frequency));

    end

    figure(1); plot(frequency, storage1); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(2); plot(frequency, storage2); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(3); plot(frequency, storage3); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(4); plot(frequency, storage4); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(5); plot(frequency, storage5); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(6); plot(frequency, storage6); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
    figure(7); plot(frequency, storage7); legend_gpr{i} = sprintf('gpr = 
%.3f', gpr);
```

```
end

hold off;

figure(1); legend(legend_gpr, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Max Gain (dB)');
figure(2); legend(legend_gpr, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Max Realized Gain (dB)');
figure(3); legend(legend_gpr, 'Location', 'best'); xlabel('Frequencyq (Hz)');
ylabel('S11 Magnitude (dB)');
figure(4); legend(legend_gpr, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('VSWR');
figure(5); legend(legend_gpr, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Return Loss');
figure(6); legend(legend_gpr, 'Location', 'best'); xlabel('Frequency (Hz)');
ylabel('Electric Field Max Magnitude');

fprintf("Total runtime was %d seconds\n", round(toc))

%% Dataset Generation for AI Models

tic;

frequency =  0.5e9 : 0.1e9 : 2e9; wavelength = freq2wavelen(frequency);

bounds = [3,15; % turns
mean(wavelength)/5, mean(wavelength)/3; % radius
mean(wavelength)/100, mean(wavelength)/50; % width
mean(wavelength)/4, mean(wavelength)/2; % spacing
mean(wavelength)/2, 2* mean(wavelength); % groundplane radius
0.5e9, 2e9]; % frequency

fsh_constant = mean(wavelength) / 500; % feedstubheight

sample_num = 100;
parameters = rand(sample_num, size(bounds, 1));

for i = 1 : size (bounds , 1)
    parameters(:,i) = parameters(:,i)*(bounds(i,2) - bounds(i,1)) +
bounds(i,1);
end

parameters(:, 1) = round(parameters(:, 1));

storage1 = zeros(sample_num,1);
storage2 = zeros(sample_num,1);
storage3 = zeros(sample_num,1);
storage4 = zeros(sample_num,1);
storage5 = zeros(sample_num,1);

for i = 1 : sample_num
    hx = helix("Turns", parameters(i,1), "Radius", parameters(i,2), "Width",
parameters(i,3), ...
    "Spacing", parameters(i,4), "FeedStubHeight", fsh_constant,
"GroundPlaneRadius", parameters(i,5));

    f = parameters(i,6);
    storage1(i) = max(max(pattern(hx,f,-180:180, -
90:90,"Type","realizedgain")));
```

```
    storage2(i) = 20*log10(abs(sparameters(hx,f).Parameters)); % s11
magnitude (dB)
    storage3(i) = -20*log10(abs(sparameters(hx,f).Parameters)); % return loss
    storage4(i) = max(max(pattern(hx,f,-180:180, -90:90,"Type","powerdb")));
    storage5(i) = max(max(pattern(hx,f,-180:180, -90:90,"Type","efield")));

    disp(i)
end

storage = [storage1, storage2, storage3, storage4, storage5];
index = randperm(sample_num);

% 0.8  training 0.2 test data division
trainindex = index(1:round(0.8*sample_num));
testindex = index(round(0.8*sample_num)+1:end);

train_parameters = parameters(trainindex,:);
test_parameters = parameters(testindex,:);

train_output = storage(trainindex, :);
test_output = storage(testindex, :);

% Normalize parameters and output data
[train_parameters, train_parameters_min, train_parameters_max] =
min_max_scale(train_parameters);
train_output = (train_output - min(train_output)) ./ (max(train_output) -
min(train_output));

test_parameters = (test_parameters - train_parameters_min) ./
(train_parameters_max - train_parameters_min);
test_output = (test_output - min(test_output)) ./ (max(test_output) -
min(test_output));

disp(toc);

%% AI Model - ANN MLP

tic;

% Train the neural network
hidden_layers = [10, 10];
net = fitnet(hidden_layers);

% Set the activation function to 'logsig' for all layers
for i = 1:length(hidden_layers)
    net.layers{i}.transferFcn = 'logsig';
end

% Train the network
net = train(net, train_parameters', train_output');

% Test the network
predicted_output = net(test_parameters');

% Calculate the mean square error
mse_value = immse(predicted_output', test_output);
disp(["Test set MSE value: " num2str(mse_value)]);

% Save the trained network
```

```
save("MLPModel.mat", "net");

disp(toc);

%% AI Model Optimize

tic;

hiddenLayerSize =
optimizableVariable('hiddenLayerSize',[10,100],'Type','integer');
trainFunction = optimizableVariable('trainFunction', {'trainscg', 'trainlm',
'trainrp', 'trainbr'},'Type','categorical');

vars = [hiddenLayerSize, trainFunction];

MinFcn = @(OptVars)minimizeMSE(OptVars, train_parameters', train_output',
test_parameters', test_output');

results = bayesopt(MinFcn, vars, 'Verbose',1, 'AcquisitionFunctionName',
'expected-improvement-plus');

optimalMSE = results.MinObjective;
optimalTrainFunction = results.XAtMinObjective.trainFunction;
optimalHiddenLayerSize = results.XAtMinObjective.hiddenLayerSize;

disp(["Optimal hidden layer size: " num2str(optimalHiddenLayerSize)]);
disp(["Optimal training function: " char(optimalTrainFunction)]);
disp(["Optimal MSE value: " num2str(optimalMSE)]);

% Retrain the network with optimal parameters
net_optimal = fitnet(optimalHiddenLayerSize, char(optimalTrainFunction));
net_optimal = train(net_optimal, train_parameters', train_output');

save('optimalNetwork.mat', 'net_optimal');

disp(toc);

%% Comparison

tic;

predicted_output1 = net(test_parameters');
predicted_output2 = net_optimal(test_parameters');

predicted_output1 = predicted_output1';
predicted_output2 = predicted_output2';

[c1, m1] = postreg(predicted_output1, test_output);
% title("Default ANN MLP Model")
disp(["Correlation for Original Network: " num2str(c1)]);
xlabel('Targets');
ylabel('Outputs');

[c2, m2] = postreg(predicted_output2, test_output);
disp(["Correlation for Original Network: " num2str(c2)]);
xlabel('Targets');
ylabel('Outputs');

disp(toc);
```

```
%% Functions

function [scaled, min_val, max_val] = min_max_scale(data)
% min_max_scale Scales the input data using min-max normalization

    min_val = min(data);
    max_val = max(data);

    scaled = (data - min_val) ./ (max_val - min_val);
end

function mse = minimizeMSE(OptVars, train_params, train_output, test_params,
test_output)
    hidden_layers = OptVars.hiddenLayerSize;
    trainFcn = char(OptVars.trainFunction);
    net = fitnet(hidden_layers, trainFcn);

    net = train(net, train_params, train_output);
    predicted_output = net(test_params);
    mse = immse(predicted_output, test_output);
end
```

RESUME

1. STUDENT PERSONAL INFORMATION

Name SURNAME	: Uygar Tolga KARA
Birth Place and Date	: Erzurum / 24.06.1996
Foreign Language	: English, German
E-mail	: uygartolgakara@gmail.com

EDUCATION STATUS

Degree	Field	School/University	Graduation Year
High School	N/A	Sınav College	2014
Bachelors	Electrical and Electronics Eng.	University of Turkish Aeronautical Assoc.	2019
Bachelors	Aviation Electrical and Electronics	Yıldız Technical University	2023 (Est.)

JOB/INTERNSHIP EXPERIENCES

Year	Firm	Duty
Feb. 2022 – June 2023	Bosch Turkey	Application and Calibration Eng.
June 2021 – Oct. 2021	Ebebek	Sales Consultant
June. 2019 – Aug. 2020	Eptim and Altınsoy	Electrical Assembly Intern