EE 492

Senior Design Project Final Report

ESTIMATION OF THE EFFECTS OF MATCHING LAYER ON WEARABLE AND IMPLANTABLE ANTENNAS USING GAUSSIAN PROCESS REGRESSION

by

Sefa Kayraklık

A report submitted for EE492 senior design project class in partial fulfillment of the requirements for the degree of Bachelor of Science (Department of Electrical and Electronics Engineering) in Boğaziçi University

Principal Investigator: Sema Dumanlı Oktar

13 February 2021

Acknowledgment

Special thanks to Bountenna members.

Abstract

Wearable and implantable devices establish body-centric wireless communications between the human body and the surroundings via wearable and implantable antennas. Since these antennas are very sensitive to their surroundings, the antennas' performances are highly affected by the presence of the body because of its highly lossy characteristics. Therefore, a *matching layer* is placed to improve the antenna performance and the quality of the communication links. Through the alteration of the permittivity and width of the matching layer, the quality of the communication links can be enhanced. The determination of the parameters of the matching layer and the antenna (which is a circular loop antenna working at the resonance frequency of 2.4 GHz.) can be accomplished by the means of simulation tools. However, in order to find the best combination of the parameter, the simulation should run for a very long time. Thus, in the project, an estimation algorithm, based on Gaussian Process Regression, to determine the best combination of the parameters is introduced in order to shorten the time for calculating the parameters.

Contents

List of Figures

List of Tables

1 Introduction

Wireless wearable and implantable devices are becoming popular within various areas such as medical, fitness, business operations, video games, etc. Wearable devices can be in the form of glasses, watches, or any kind of tools that are worn. These devices are used to track the user's movement, location, vital signs, calorie expenditure, sleep, and some other specific information. Implantable devices are more precise and usually require surgery; however, they can obtain information that cannot be accessible to sensors externally placed to the body. In the medical field, wearable and implantable devices have applications of monitoring, diagnosis, and treatment of various diseases.

Body-centric wireless communications (BCWC) are established using wearable and implantable devices which provide a wireless communication link between the human body and the surroundings via wearable and implantable antennas. There exist mainly three different communication types in BCWC such as *on-body* communications, which occur between on-body devices, *in-body* communications, which are formed within implanted devices, and *off-body* communications, which occur between on-body and off-body devices [1]. The Figure 1 shows a schematic of the mentioned communication types.

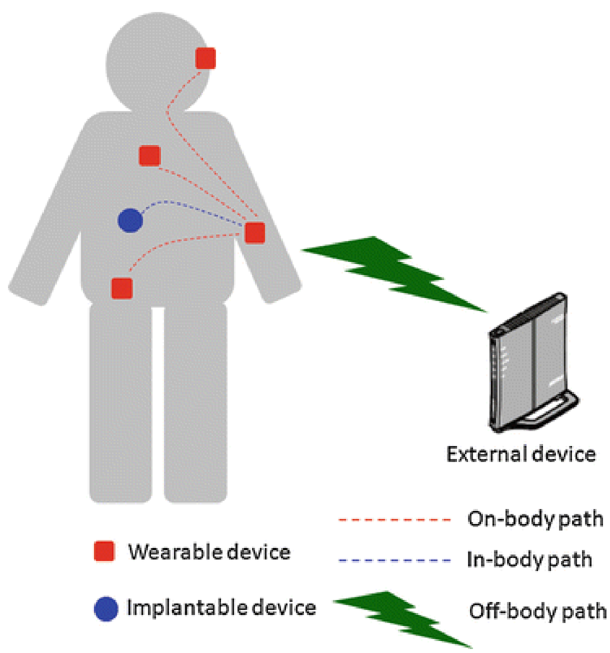


Figure 1: A schematic of body-centric wireless communications [1]

The antennas used in these devices are very sensitive to their surroundings. The properties of the implanted antenna such as average transmitted power, return loss, resonance frequency, bandwidth, etc are highly impacted by the presence of the body due to the fact that the human body is formed by highly lossy materials for radio frequency propagation [2]. These negative effects are mainly stemmed from path losses, which are caused by the impedance mismatch between the propagation environments (air and the body), near field losses and reflection. These factors result in inefficient power transmission of the radio wave in the in-body communications and the on-body communications. Therefore, in order to improve the quality of the in-body and on-body communication links, a *matching layer* can be placed. By altering the permittivity and width of the matching layer, the quality of the in-body and on-body communication links can be enhanced.

1.1 Objectives

The determination of the matching layer parameters is accomplished by the means of simulation tools such as ANSYS HFSS. Also, where to place the antenna is a part of the design problem since the permittivity of the human body is not constant and it affects the performance of the antenna. Thus, in order to decide the optimum matching layer parameters and the location of the antenna, the simulation should run for a very long time.

The antenna which is simulated is determined as a circular loop antenna that works at the resonance frequency of 2.4 GHz. So, the first objective is to design a circular loop antenna in the HFSS environment and to simulate in order to generate the training data for an estimation model which is constructed by a Gaussian Process Regression method.

The aim of the project is to build a Gaussian Process Regression model that can estimate the antenna's return loss and the transmitted average power in terms of the permittivity and width of the matching layer, the permittivity of the human body, and the antenna dimensions.

1.2 Approach and Methodology

As a first step of the project, the circular loop antenna theory will be examined and a circular loop antenna at a resonance frequency of 2.4GHz will be designed via HFSS. After obtaining the antenna, the literature about the machine learning methods used in the field of wearable and implantable antennas will be investigated. The knowledge of how the Gaussian Process Regression is constructed and which models are used will be obtained. After gaining an insight into the approach of the regression model, a dataset should be generated in order to train and test the constructed model. The dataset will be generated by running simulations of the designed antenna in the HFSS with some predetermined parameters of matching layers and the human body. Finally, with an adequate dataset, the well-constructed model will be trained and tested.

2 Circular Loop Antenna

Loop antennas are one of the basic and cheap antenna types, also they are not complicated in production and analysis [3]. The loop antennas can be in the shape of rectangle, square, triangle, ellipse, circle and some other configurations. In the project, a circular shape is considered and the design procedure and the formulations of the textbook of Antenna Theory Analysis and Design [3] are followed to obtain a circular loop antenna whose resonance frequency is at around 2.4 GHz.

2.1 Design Procedure of the Antenna

The circular loop antenna is designed in HFSS. First, a circle with a radius of *wire_rad* is placed at a distance of *torus_rad* from the origin. Then, it is swept around the origin by the amount of

$$\frac{2\pi torus\_rad}{2\pi torus\_rad + port\_gap} * 360$$

degrees. After obtaining the torus with the given *port_gap*, a line is drawn with the length of 2*wire_rad* across the center of the wire. Then it is also swept around the origin by the amount of

$$\frac{port\_gap}{2\pi torus\_rad + port\_gap} * 360$$

degrees in order to obtain the excitation port.

The circumference of the circular loop antenna where the self-resonance occurs is selected. By referring the input impedance of circular loop antennas, which is shown in the Figure 2, and choosing $\Omega = 12$, Lommel–Weber function [3], the circumference of the loop is approximately 1.089λ.

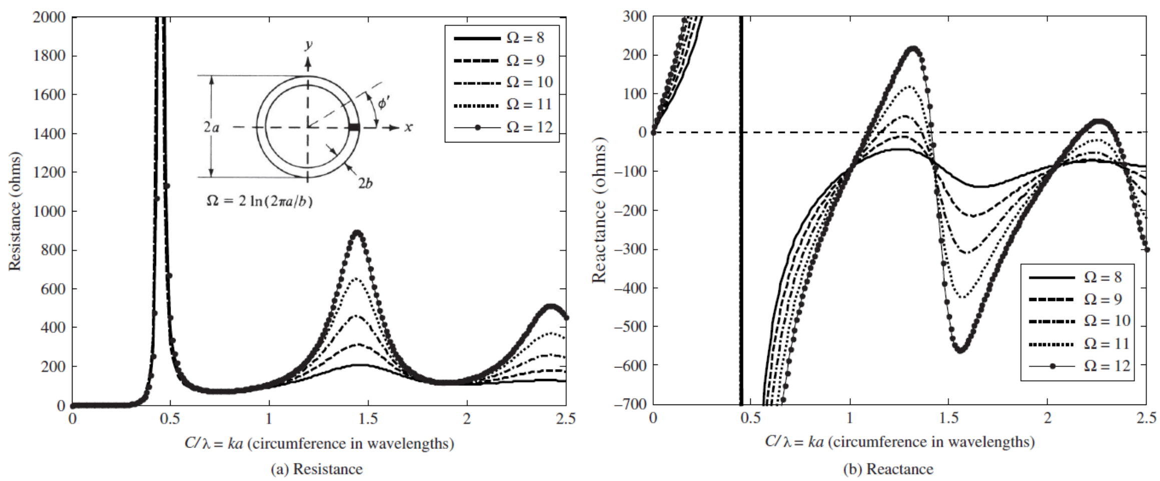


Figure 2: Input impedance of circular loop antennas[3]

So, the *torus_rad* can be found at the frequency of $2.4GHz$ as following:

$$r_{torus} = \frac{1}{2\pi} 1.089 \frac{c}{f} = \frac{1.089}{2\pi} \frac{3 * 10^8}{2.4 * 10^9} \approx 2.15cm$$

The radius of the wire is obtained as following expression [3]:

$$\Omega = 12 = 2ln\left(\frac{2\pi r_{torus}}{r_{wire}}\right)$$
$$\rightarrow r_{wire} = 2\pi \frac{2.15cm}{e^6} \approx 0.0335cm$$

The *port_gap* of the antenna is selected as a value of $0.0335cm$, comparable to the *wire_rad*. The evaluated values of the antenna dimensions are implemented in the HFSS environment.

The Figure 3 shows the constructed model in HFSS.

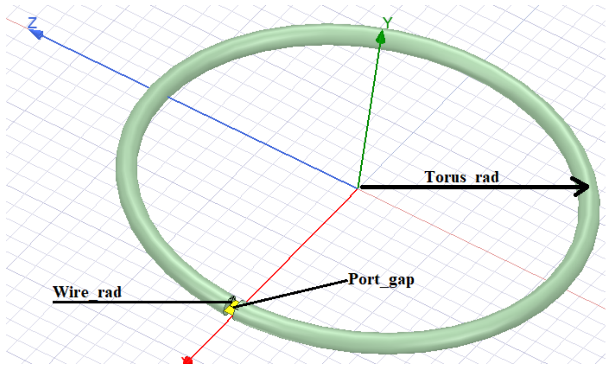


Figure 3: Model in HFSS

The yellow plane is configured as a lumped port and the material of the solid loop is selected as a perfect electric conductor (PEC).

2.1.1 Simulation Results

The simulation of the model which is proposed in the previous section with *torus_rad* of $2.15cm$, *wire_rad* of $0.0335cm$ and *port_gap* of $0.0335cm$ is run, and the 3D and 2D radiation pattern of the antenna are given in the Figure 4.

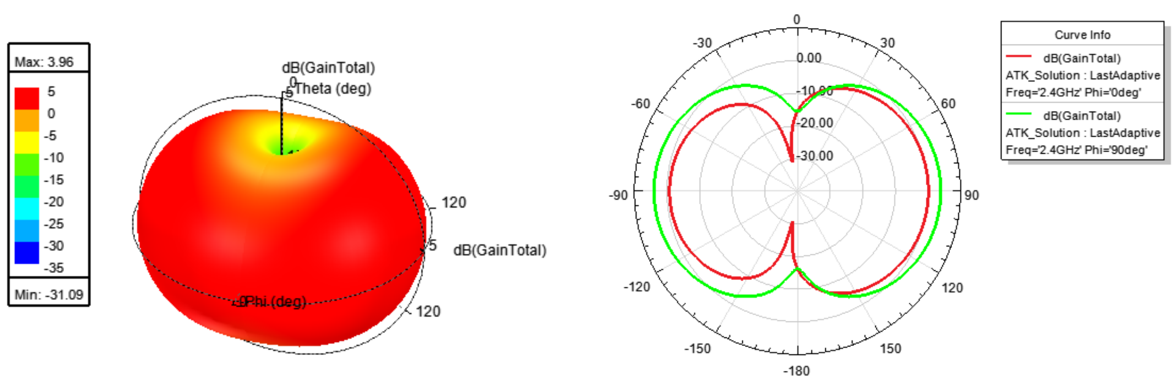


Figure 4: The gain plots of the antenna

The return loss graph of the simulated antenna is shown in the Figure 5. These values are not as expected to be since the return loss curve should be below at least $-10dB$ at the resonance frequency of $2.4GHz$.

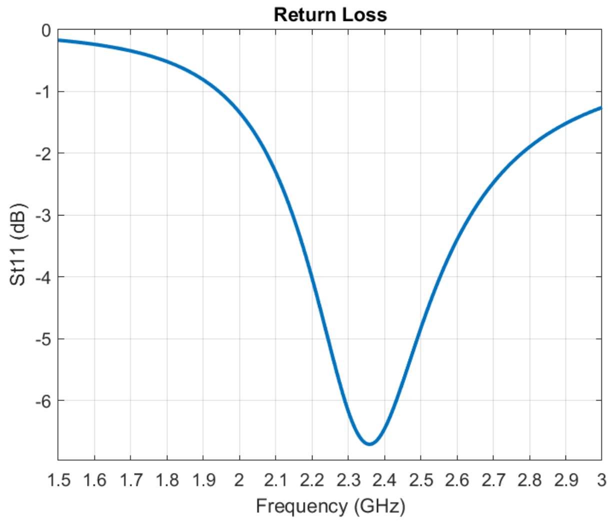


Figure 5: The return loss graph of the antenna

The effects of changing *torus_rad*, *wire_rad*, and *port_gap* are analyzed:

• **Torus Radius:**

The torus radius of the antenna is altered from $1.8cm$ to $2.4cm$ by increments of $0.1cm$ while setting the wire radius and port gap as $0.1cm$, and the Figure 6 shows the return loss of the selected parameters.

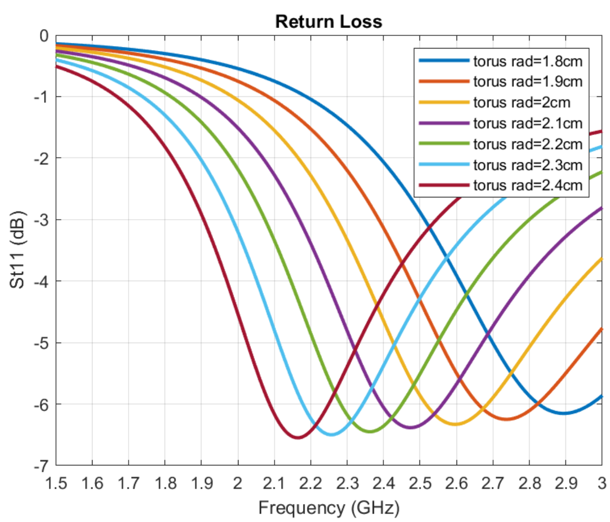


Figure 6: The effects of changing torus radius on the return loss

So, it can be observed from the above figure that the implemented design behaves like an antenna since as its dimension increases, its resonance frequency becomes lower.

• **Wire Radius:**

Two sets of parameters are simulated to observe the effects of varying the wire radius. The first one is to change the wire radius from $0.01cm$ to $0.05cm$ by the increments of $0.01cm$ while setting the torus radius as $2.15cm$ and port gap as $0.03cm$, and the Figure 7 shows the obtained results. The second one is to change the wire radius from $0.1cm$ to 0.5 by the increments of $0.1cm$ while setting the torus radius as $2.15cm$ and port gap as $0.1cm$, and the Figure 7 shows the obtained results.

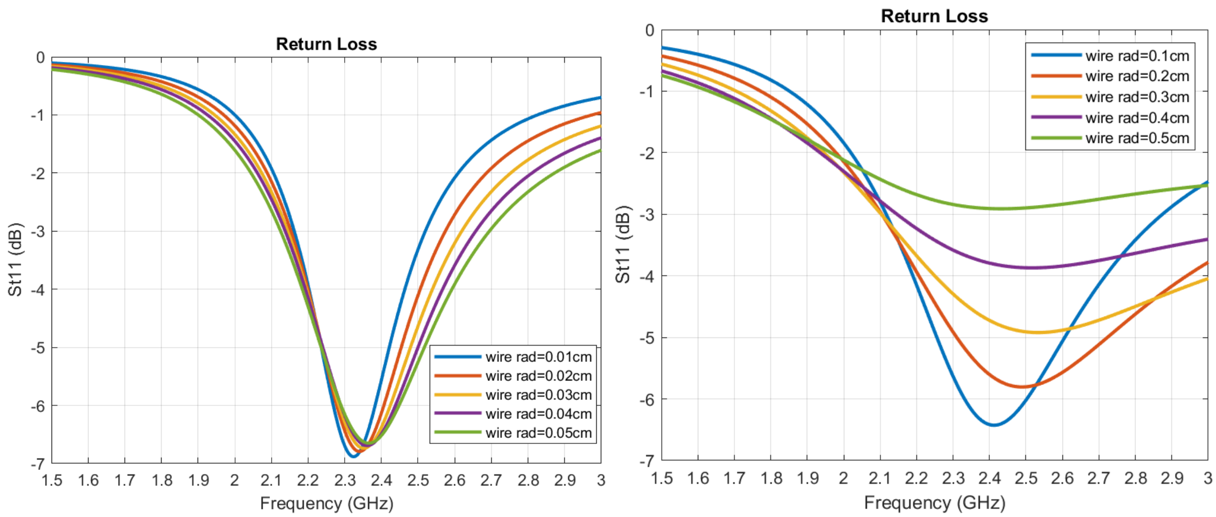


Figure 7: The effects of smaller and larger increments of the wire radius on return loss, respectively

In fact, it is expected that the change in the wire radius will improve the return loss results; however, the estimated impacts of altering the wire radius don't occur in the above findings.

- **Port Gap:**

Similarly, two sets of parameters are simulated to observe the effects of varying the port gap. The first one is to set the port gap $0.03cm$, $0.05cm$, and $0.1cm$ while setting the torus radius as $2.15cm$ and wire radius as $0.03cm$, and the Figure 8 shows the obtained results. The second one is to change the wire radius from $0.1cm$ to $0.5cm$ by increments of $0.1cm$ while setting the torus radius as $2.15cm$ and port gap as $0.1cm$, and the Figure 8 shows the obtained results.

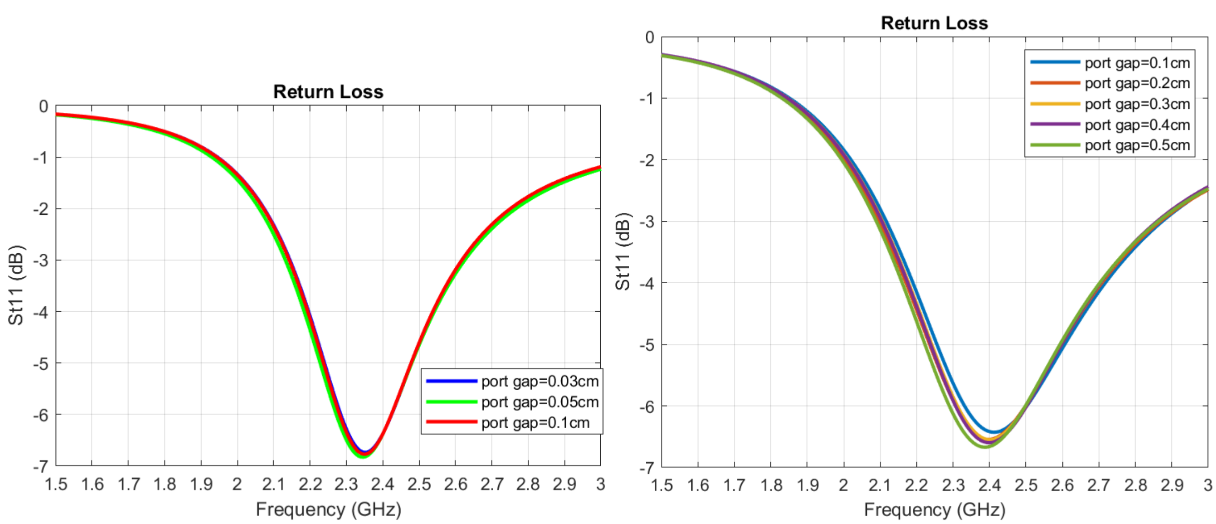


Figure 8: The effects of smaller and larger increments of the port gap on return loss, respectively

2.1.2 Conclusion

An attempt of designing a circular loop antenna that works at the frequency of 2.4Ghz is made; unfortunately, the expected results cannot be obtained by altering the wire radius of the antenna. In fact, the change in the torus radius affects the results as expected. In order to match the antenna, a change in its excitation can be made rather than altering its dimensions. Therefore, a literature review about matching a circular loop antenna is done in order to observe different implementations of the antenna.

2.2 Circular Loop Antenna Examples

In the implementation of a circular loop antenna, the input impedance matching to 50Ω is a challenge for antenna designers; therefore, they overcome this issue by taking different approaches to the antenna design.

2.2.1 Circular Loop Antenna with 8 Feed Points

The proposed antenna [6] consists of a conductor strip loop of inner radius $r = 19.09mm$ whose strip width is $w = 1.5mm$ and thickness of $h = 12mm$ which is shown in the Figure 9.

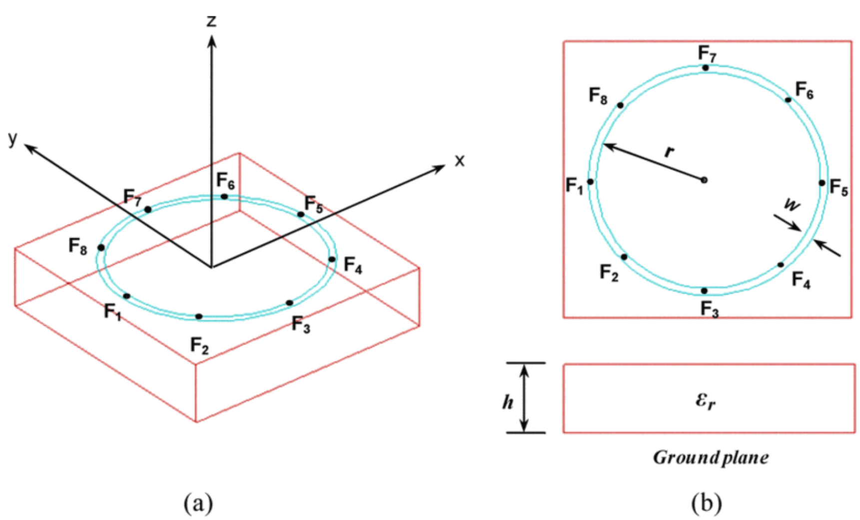


Figure 9: Circular loop antenna with 8 feeds: (a) Perspective view and (b) Top and side view

8 feeding points F_1, F_2, F_3, F_4,F_5, F_6, F_7, and F_8, which are placed over the circular loop with an angular spacing of 45°, feed the antenna. The antenna performance is the same for all the feed locations because of the antenna's symmetrical shape, except for the directions of the beam. Hence, only one feed configuration, F_1, is analyzed assuming that the other feed configurations give the same results.

The antenna is analyzed for the feed configuration of F_1 and Figure 10 shows its return loss for 50Ω input impedance. The resonance frequency of the antenna is $f_0 = 4.4GHz$. As shown in Figure 10, the antenna shows a -10 dB bandwidth of $550MHz$ range from $4.2GHz$ to $4.75GHz$.

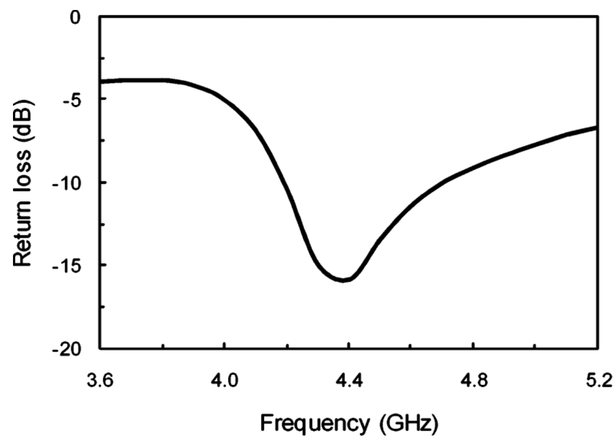


Figure 10: Return Loss

2.2.2 Partitioned Circular Loop Antenna with Five Capacitive Elements

The loop of the proposed antenna [7] is composed of partitioned into multiple segments and capacitive elements at chosen locations in order to decrease phase variations in the current flow and hence to enhance the radiation efficiency. A total of five capacitors are placed to obtain stable current flow. A thin-wire circular loop antenna of radius $r = 7.0mm$, circumference $C \approx 44.0mm$ $(0.85\lambda_0)$ and wire segments of radius $a = 0.5mm$ is designed and its resonance frequency is 5.8 GHz. 24 nodes consisting of 24 wire segments form the antenna geometry, as shown in Figure 11.

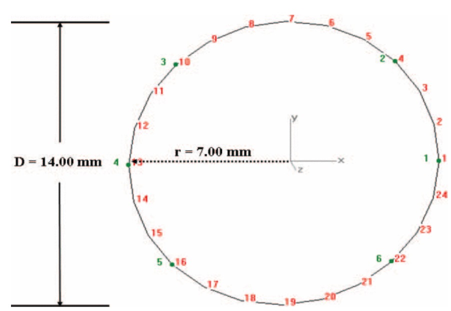


Figure 11: Thin wire model of partitioned circular loop antenna with wire nodes (red) and lumped series ports (green) indicated

Five capacitive elements are placed at the following node locations to cancel out the net inductance of the wire loop: 0.086 pF (nodes 4 and 22), 0.042 pF (nodes 10 and 16), and 0.047 pF (node 13) [7]. The simulation of the antenna from 1 to 10 GHz is run, resulting in an input impedance of $49.3 - j0.05\Omega$, and the return loss graph is shown in Figure 12.

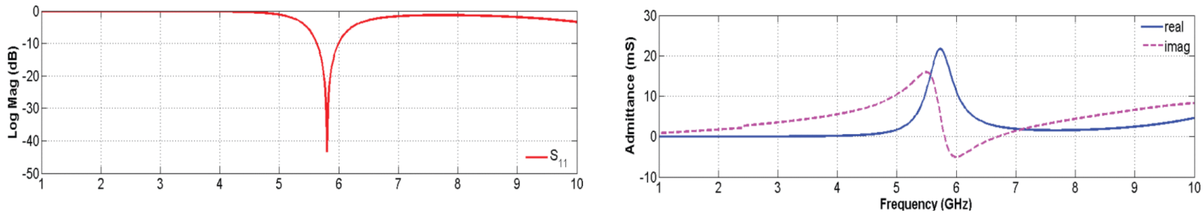


Figure 12: Return loss and input admittance for the thin wire partitioned circular loop antenna

With the insight of the above circular loop antennas, a matching network will be connected to the designed antenna port in order to match the input impedance with the 50Ω.

2.3 Planar Circular Loop Antenna with Matching Network

As concluded in the previous parts, a circular loop antenna could not be matched by altering its dimensions since, in the region of its self-resonance, the antenna has an input impedance of almost 150Ω. Therefore, in order to match its input impedance to 50Ω, a matching network consisting of a capacitor and an inductor is needed.

Detailed information about matching networks is given in the Appendix-A.

In the circular loop antenna case, the first step is achieved by connecting a capacitor in parallel and the second step is to connect an inductor in series since the antenna's input impedance is placed to the right bottom of the Smith chart. The steps will be further discussed in the following part.

2.3.1 Antenna Design with Matching Network

The antenna is designed with a similar approach in the previous section. The constructed antenna model in HFSS is shown in the Figure 13.

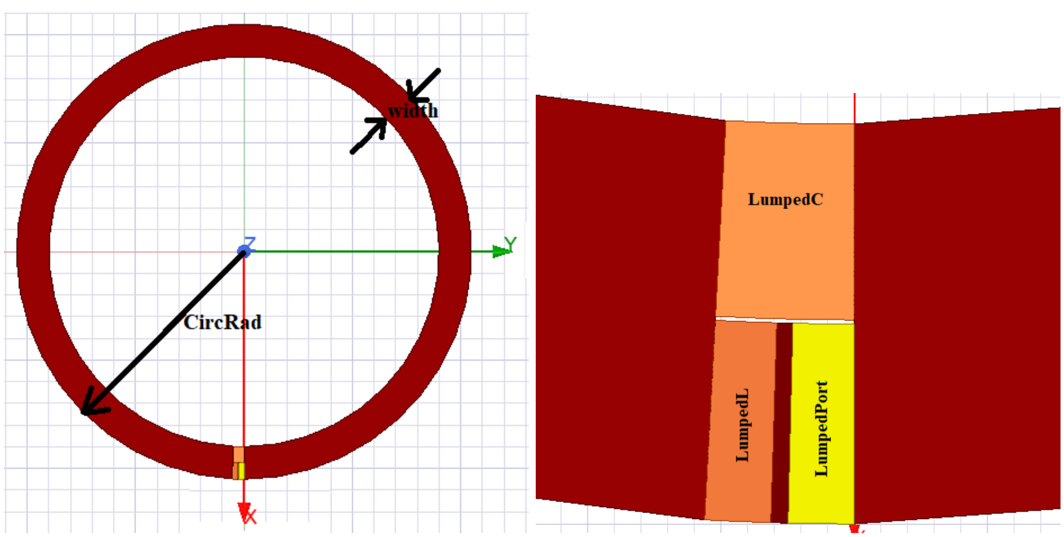
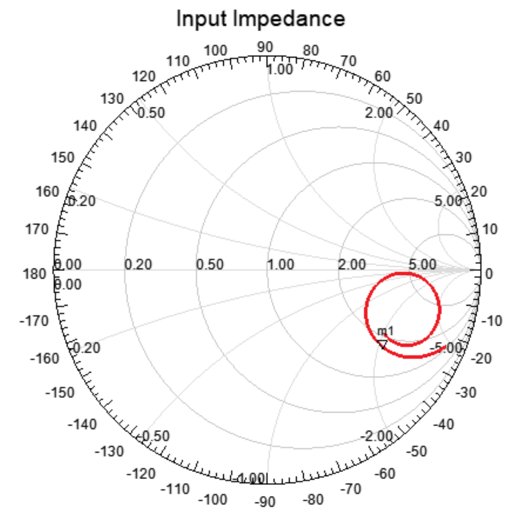


Figure 13: Planar circular loop antenna model with a matching network in HFSS

The circumference and the width of the circular loop antenna is adopted from the previous section; so the circular loop radius is $2.1cm$, the loop width is $0.3cm$ and the port width is $0.1cm$.

With the above parameters, the HFSS model (before adding a capacitor and an inductor) is simulated between the frequencies of $2Ghz$ and $5Ghz$; and the obtained input impedance of the antenna is plotted on the Smith chart which is shown in the Figure 14.



Name	Freq	Ang	Mag	RX
m1	2.4100	-33.6782	0.6563	1.6821 - 2.1506i

Figure 14: The input impedance without a matching network

The antenna has an input impedance of $50*(1.6821-2.1506i)\Omega$ at $2.41GHz$. So, in order to move the impedance to the center of the chart (matching to 50Ω), a shunt capacitor and a series inductance are needed.

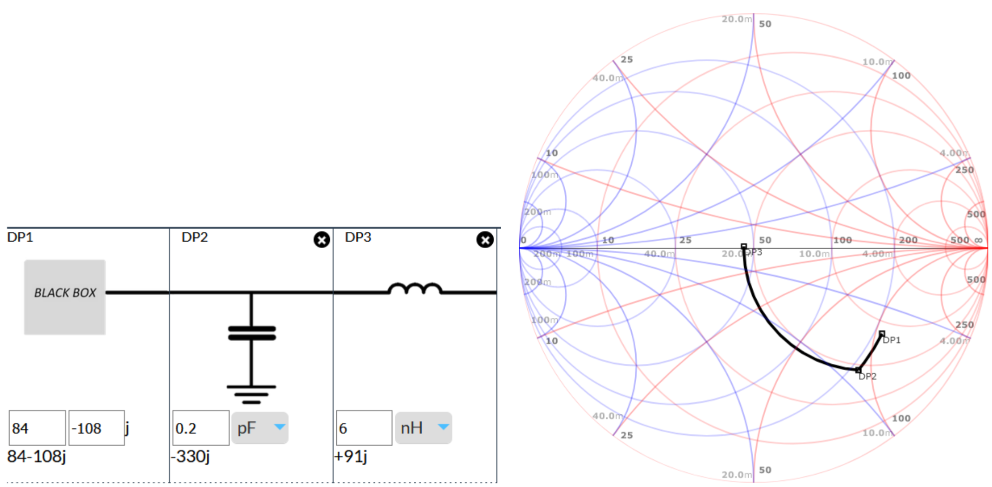


Figure 15: The circuit of antenna and matching network, and the effects of adding the matching network, respectively

In order to calculate the values of capacitance and inductance, an online Smith Chart tool [8] is used. The inspected circuit is shown in the Figure 15. The black box has the impedance of the antenna.[1] Also, the effects of adding the specified capacitance and inductance values on the impedance is shown on the Smith Chart in the Figure 15.

The HFSS model with the lumped capacitance of $0.2pF$ in parallel connection and the lumped inductance of $6nH$ in series connection is simulated between the frequencies of $2GHz$ and $3GHz$, and the obtained input impedance of the antenna is plotted on the Smith chart which is shown in the Figure 16.

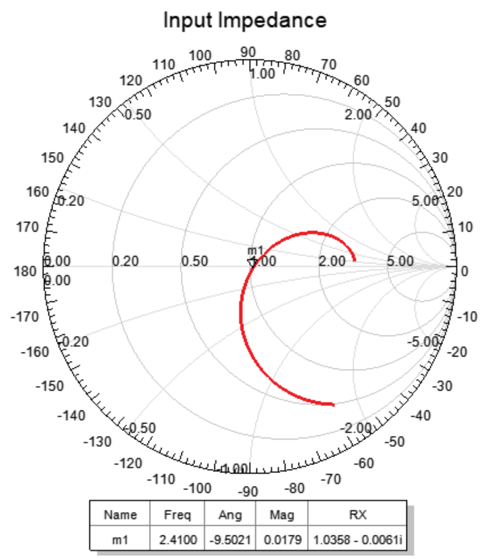


Figure 16: The input impedance of the antenna with the matching network

The return loss graph and the radiation pattern of the antenna with the matching network are given in the Figure 17, respectively.

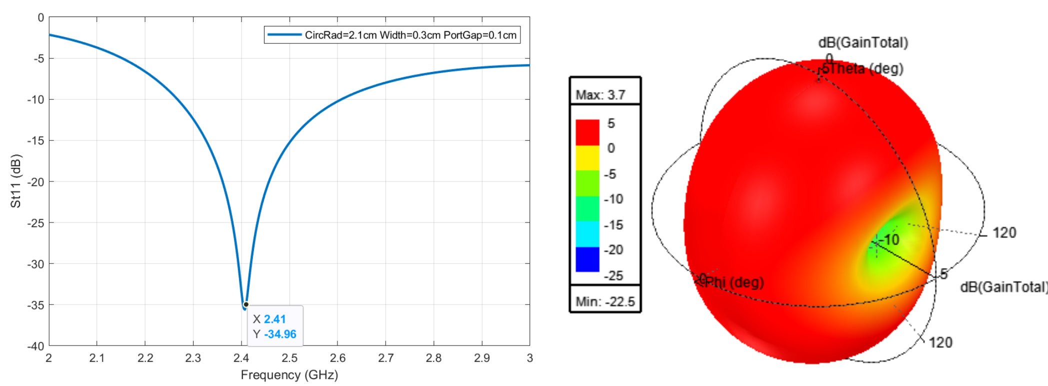


Figure 17: The return loss and radiation pattern of the antenna, respectively

[1]Note impedance is looking towards the BLACK BOX.

2.3.2 Conclusion

The planar circular loop antenna with $2.1cm$ radius, $0.3cm$ width and $0.1cm$ port gap is matched to 50Ω with a matching network of a parallel capacitance of $0.2pF$, and a series inductance of $6nH$. So, with the matching network, the antenna can operate with an approximately 35dB return loss at the specified frequency of $2.4GHz$. However, there exists a drawback of having a matching network, which is whenever the dimensions of the antenna are altered, new values of capacitance and inductance should be calculated and connected.

3 Antenna in the Presence of Tissue

As concluded in the previous part, a planar circular loop antenna can be matched to 50Ω by using a matching network consisting of a capacitor and/or an inductor. In the following implementations, the designed antenna is simulated and matched in two different setups: the presence of only tissue layer, and tissue and matching layers.

In order to match the antenna, the antenna without the matching network is first simulated to determine the input impedance; then, with the calculated values of capacitance and inductance at the frequency of 2.4GHz, the input impedance will be shifted to the center of the Smith chart (50Ω).

3.1 Antenna with Tissue Layer

The constructed simulation setup is shown in the Figure 18. The light green cube is the tissue layer whose permittivity is 52.17 and conductivity is $1.71S/m$, and the dark red structure is the designed antenna.

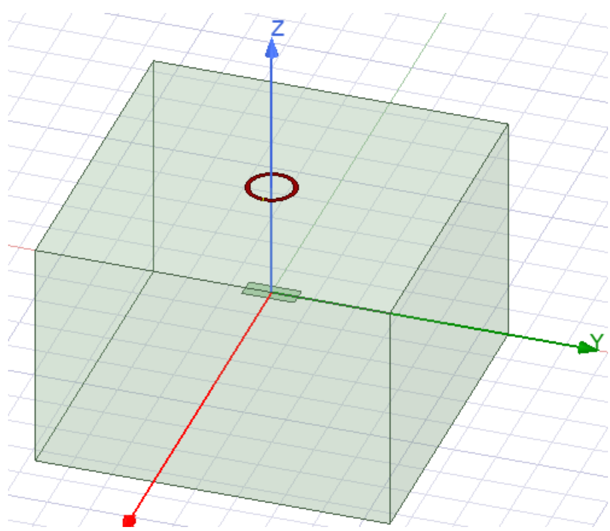


Figure 18: The simulation setup with tissue

3.1.1 Antenna without Matching Network

The HFSS model is simulated with the parameter of circular loop radius of from $1.2cm$ to $2.4cm$ by the increments of $0.3cm$, wire width of $0.3cm$ and port gap of $0.1cm$ is simulated between the frequencies of $1GHz$ and $3.GHz$. Since the resonance region could not be found, the model is again simulated with the parameter of circular loop radius of $1.2cm$ and $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$ is simulated between the frequencies of $1GHz$ and $3.GHz$. The obtained return loss graphs are shown in the Figure 19, respectively.

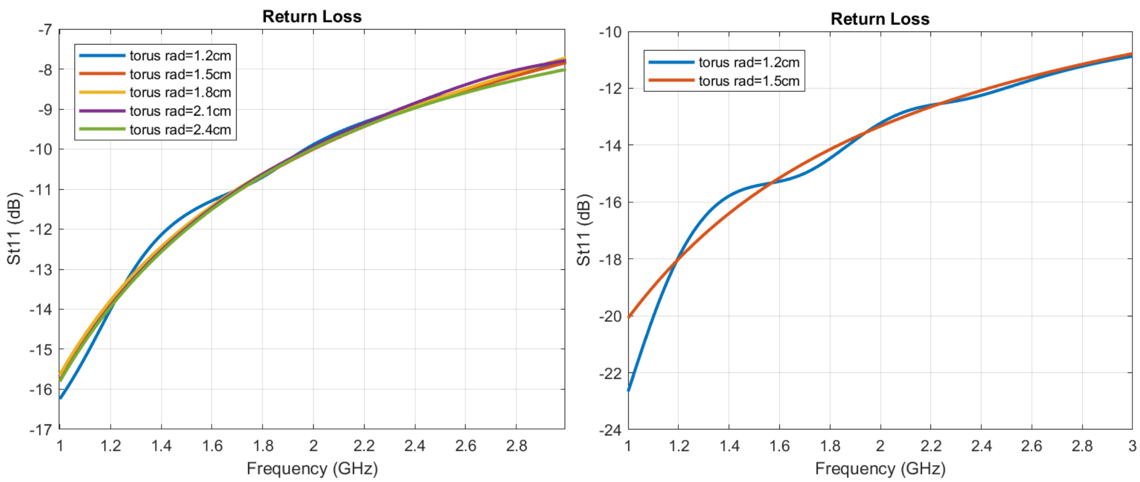
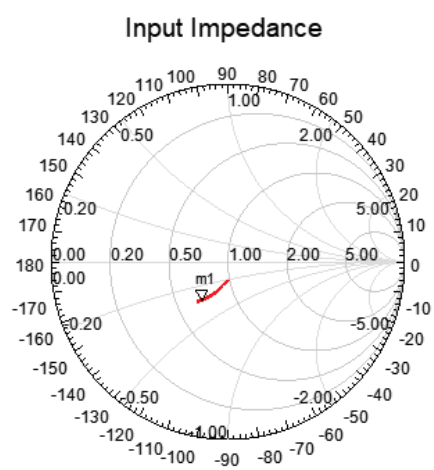


Figure 19: The return loss with tissue layer and wire width of $0.3cm$ and $0.2cm$, respectively



Name	Freq	Ang	Mag	RX
m1	2.4000	-122.6732	0.2490	0.7049 - 0.3149i

Figure 20: The input impedance with tissue layer and circular loop radius of $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$

Actually, due to the *not enough memory error*, the size of the antenna could not be furthermore shrunk; therefore, the resonance region could not be determined. So, the simulation proceeds to the matching 50Ω via the matching network with the best-obtained parameters of circular loop radius of $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$.

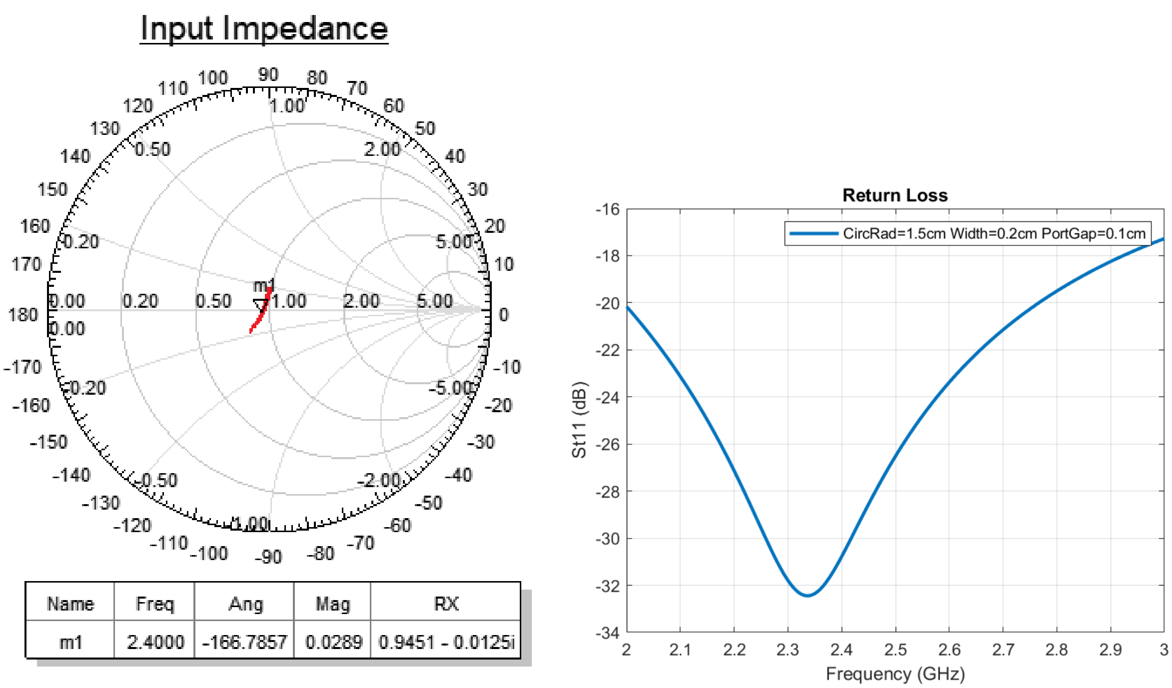
In order to construct the matching network, the input impedance of the antenna with the above-mentioned parameters is shown in the Figure 20. So, to match the impedance to 50Ω, a series capacitor to shift to the $20mS$ circle and a parallel inductor to shift to the center are needed.

3.1.2 Antenna with Matching Network

The values of the external components are calculated at the frequency of 2.4GHz via an online Smith Chart tool [8] as $8pF$ of series capacitance and $4nH$ of parallel inductance. Since the matching network is changed from the previous project, the antenna port is shown in the Figure 22.

The HFSS model with the determined parameters in the previous section and the above determined values is simulated between the frequencies of $2GHz$ and $3.GHz$. The obtained input impedance and return loss graphs are shown in the Figure 21, respectively.

Since the simulation with the further smaller dimensions could not be done due to the lack of memory, the best-achieved results are shown above.



Name	Freq	Ang	Mag	RX
m1	2.4000	-166.7857	0.0289	0.9451 - 0.0125i

Figure 21: The input impedance and return loss with tissue layer, circular loop radius of $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$ and matching network, respectively

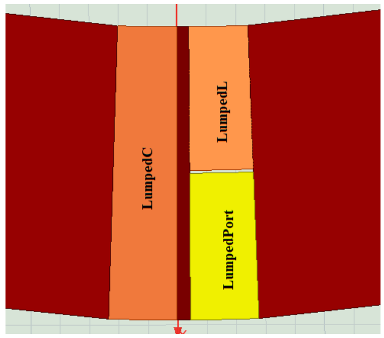


Figure 22: The antenna port with the matching network

3.2 Antenna with Tissue and Matching Layer

The same procedure is repeated in the presence of a matching layer whose permittivity is 22.

The constructed simulation setup is shown in the Figure 23. The light green cube is the tissue layer, the yellow one is the matching layer and the dark red structure is the designed antenna.

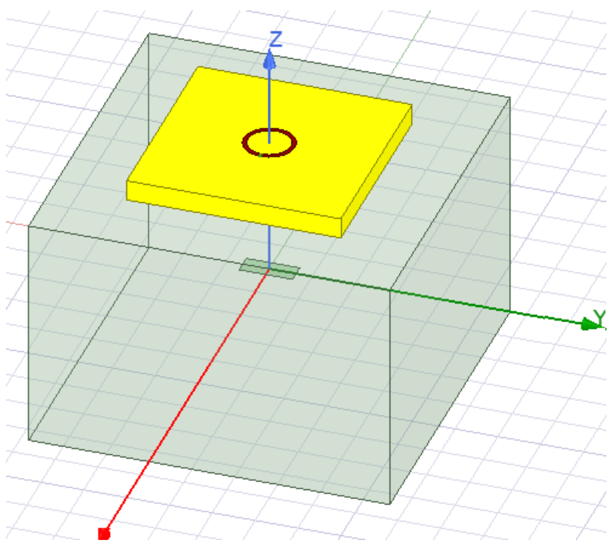


Figure 23: The simulation setup with tissue and matching layers

3.2.1 Antenna without Matching Network

The HFSS model is simulated with the parameter of circular loop radius of from $1.2cm$ to $2.1cm$ by the increments of $0.3cm$, wire width of $0.3cm$ and port gap of $0.1cm$ is simulated between the frequencies of $2GHz$ and $3.GHz$. The obtained return loss is shown in the first half of the Figure 24. It can be seen from the above figure that the resonance region for 2.4GHz can be acquired with the parameters of circular loop radius of $1.2cm$, wire width of $0.3cm$ and port gap of $0.1cm$. Also, the simulation is run for the wire width of $0.2cm$, the obtained return loss is shown in the second half of the Figure 24.

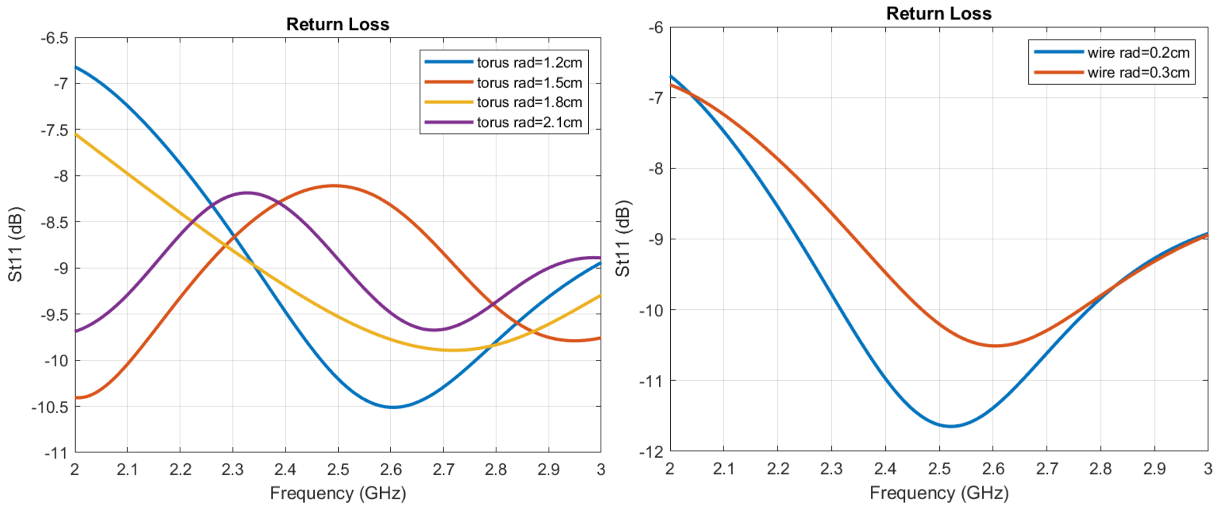
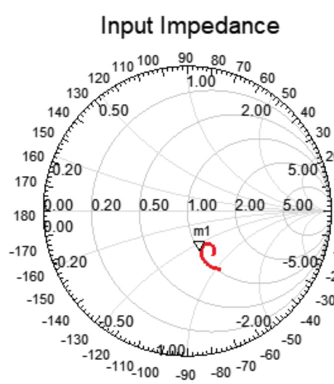


Figure 24: The return loss with tissue and matching layers and wire width of $0.3cm$, circular loop radius of $1.2cm$; respectively

So, the simulation proceeds to the matching 50Ω via matching network with the best obtained parameters of circular loop radius of $1.2cm$, wire width of $0.2cm$ and port gap of $0.1cm$.

In order to construct the matching network, the input impedance of the antenna with the above-mentioned parameters is shown in the Figure 25.



Name	Freq	Ang	Mag	RX
m1	2.4000	-69.6992	0.2827	1.0411 - 0.6001i

Figure 25: The input impedance with tissue and matching layers, circular loop radius of $1.2cm$, wire width of $0.2cm$ and port gap of $0.1cm$

So, to match the impedance to 50Ω, a parallel capacitance to shift to the 50Ω circle and a series inductor to shift to the center is needed.

3.2.2 Antenna with Matching Network

The values of the external components are calculated at the frequency of 2.4GHz via an online Smith Chart tool [8] as $0.1pF$ of parallel capacitance and $2nH$ of series inductance. And the antenna port is shown in the Figure 26.

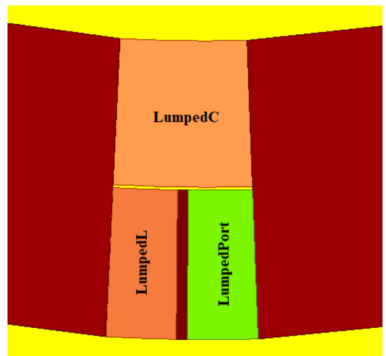


Figure 26: The antenna port with the matching network

The HFSS model with the determined parameter in the previous section and the above determined values is simulated between the frequencies of $2GHz$ and $3.GHz$. The obtained input impedance and return loss are shown in the Figure 27, respectively.

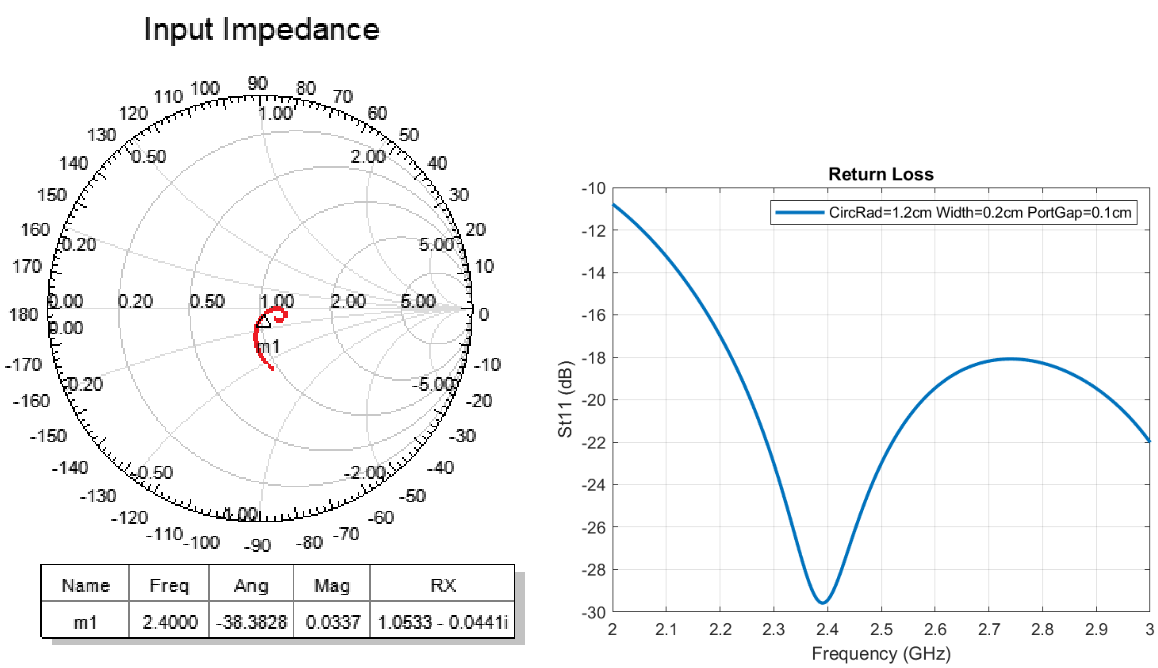


Figure 27: The input impedance and return loss with tissue and matching layers, circular loop radius of $1.2cm$, wire width of $0.2cm$ and port gap of $0.1cm$ and matching network, respectively

3.3 Delivered Average Power Calculation through Poynting Vector

As concluded in the previous section, the planar circular loop antenna, working at the frequency of 2.4 GHz, in the presence of tissue and matching layers, is matched to the impedance of 50Ω with a matching network circuit. The matching network connections and the values of the component are modified according to the input impedance of the antenna.

In order to analyze the effect of adding a matching layer between the tissue and the antenna, the average power delivered to the center of the tissue layer is considered. It is calculated as first, determining the total power observed by a defined rectangular surface on the center of the tissue and then, dividing the computed total power to the area of the rectangular:

$$P_{ave} = \frac{\oiint_S \mathscr{W} \cdot d\mathbf{S}}{\oiint_S dS}$$

, where $\mathscr{W}$ is the Poynting vector of the antenna, and S is the defined rectangular surface on the center of the tissue.

3.3.1 Simulation with Only Tissue Layer

Firstly, the planar circular loop antenna, whose resonance frequency is $2.4GHz$, is optimized without using a matching network. Since in the previous section, the antenna was not further shrunk, the resonance frequency of $2.4GHz$ could not be reached without using a matching network. However, in this section, the antenna is made further smaller (its radius becomes $0.5cm$ which is approximately 25% of the actual dimension).

Therefore, the simulation of the loop antenna of loop radius of $0.5cm$ and $0.8cm$, wire width of $0.1cm$ and port gap of $0.1cm$ in the presence of only tissue layer is run and the corresponding return loss graphs are shown in the Figure 28.

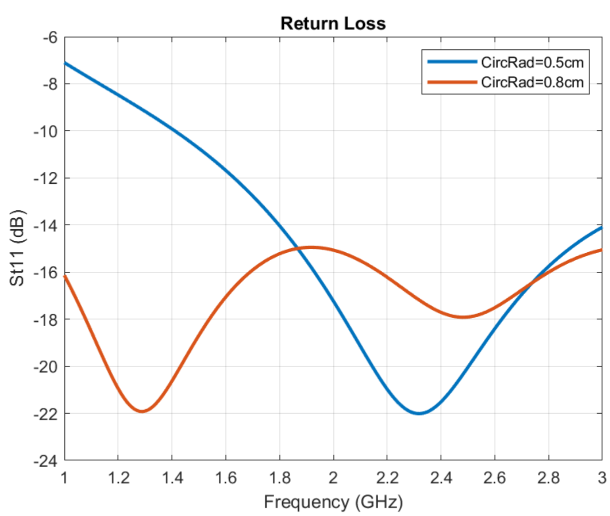


Figure 28: The return loss with tissue layer and wire width of $0.1cm$

The calculated average power of the above antenna with the dimensions of loop radius of $0.5cm$, wire width of $0.1cm$ and port gap of $0.1cm$ is $\mathbf{0.0064W/m^2}$.

The dimensions of the circular loop antenna with a matching network, which was designed in the previous section, were the circular loop radius of $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$, and the matching network consisted of $8pF$ of series capacitance and $4nH$ of parallel inductance. Its return loss was shown in the Figure 21. The calculated average power of the mentioned antenna with the dimensions of loop radius of $1.5cm$, wire width of $0.2cm$ and port gap of $0.1cm$ is $\mathbf{0.0020W/m^2}$.

3.3.2 Simulation with Both Tissue and Matching Layer

Firstly, the planar circular loop antenna in the presence of both tissue and matching layer is optimized without using a matching network. As in the previous section, various parameters of antenna dimensions are considered in the simulation and the resonance frequency of $2.4GHz$ is obtained in the antenna dimensions of loop radius of $0.7cm$, wire width of $0.1cm$ and port gap of $0.1cm$.

Therefore, the simulation of the loop antenna of loop radius of $1.2cm$, $0.8cm$ and $0.7cm$, wire width of $0.1cm$ and port gap of $0.1cm$ in the presence of both tissue and matching layer is run and the corresponding return loss graphs are shown in the Figure 29.

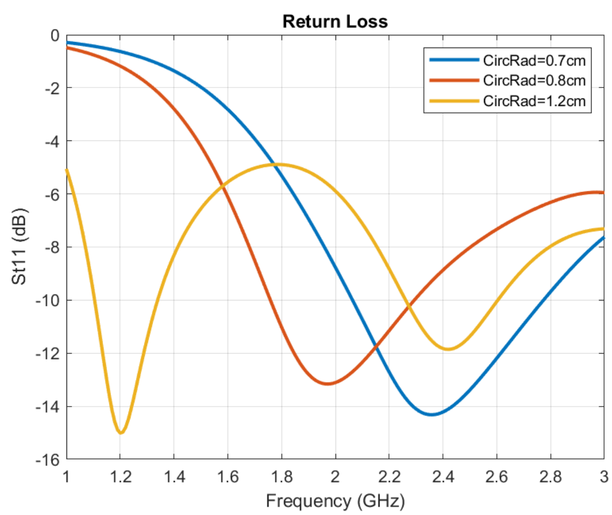


Figure 29: The return loss with tissue and matching layers, and wire width of $0.1cm$

The calculated average power of the above antenna with the dimensions of loop radius of $0.7cm$, wire width of $0.1cm$ and port gap of $0.1cm$ is $\mathbf{0.0103W/m^2}$.

The dimensions of the circular loop antenna with a matching network, which was designed in the previous section, were the circular loop radius of $1.2cm$, wire width of $0.2cm$ and port gap of $0.1cm$, and the matching network consisted of $0.1pF$ of parallel capacitance and $2nH$ of series inductance. Its return loss was shown in the Figure 27. The calculated average power of the above antenna with the dimensions of loop radius of $1.2cm$, wire width of $0.2cm$ and port gap of $0.1cm$ is $\mathbf{0.0059W/m^2}$.

3.3.3 Results

The Table 1 shows all the average power values which are calculated in the previous part.

Table 1: Delivered average power of the planar circular loop antenna

		Circle Rad (cm)	Wire Width (cm)	Port Gap (cm)	Value (W/m^2)
Matching Layer	Matching Network	1.2	0.2	0.1	0.0059
	No Matching Network	1.2 0.7	0.2 0.1	0.1 0.1	0.0054 0.0103
No Matching Layer	Matching Network	1.5	0.2	0.1	0.0020
	No Matching Network	1.5 0.5	0.2 0.1	0.1 0.1	0.0019 0.0064

There exist three observations to point out from the table:

- The matching layer increases the delivered average power by almost two times.
- The matching network does not affect the average power as the matching layer does and it increases the average power almost 10 percent.
- Making the antenna much smaller also increases the delivered average power almost two times.

3.3.4 Conclusion

As observed in the above table, the contribution of using a matching network in the presence of tissue layer is so small in comparison to the placing matching layer. Furthermore, for each simulation parameter, the values of the components in the matching network should be calculated separately. Therefore, in order to simplify the process of generating a dataset for the estimation model, the matching network circuit is not included in the simulation model.

4 Dataset Generation

A dataset that will be used in training a model to estimate the antenna's return loss and delivered average power is generated using HFSS simulation program.

4.1 Simulation Setup

As concluded in the previous section, the simulations are run with tissue and matching layers excluding the matching network circuit in the antenna's port. The simulation setup can be seen in the Figure 23. The simulation model parameters are given in the Table 2, and 3.

Table 2: Simulation Parameters for the Antenna

Antenna Dimensions	Values (cm)					
circle radius	0.6	0.7	0.8	0.9	1.0	1.1
circle width	0.1	0.1	0.1	0.2	0.2	0.2
port gap	0.1					

Table 3: Simulation Parameters for the Tissue and Matching Layer

Tissue and Matching Layer	Values			
Tissue permitivity	20	30	40	50
Matching layer permitivity	[1 7 14]	[1 8 16 23]	[1 9 17 24 32]	[1 9 17 26 34 42]
Tissue dimensions	15× 15×10 cm			
Matching layer dimensions	5×5×0.5 cm		5×5×1 cm	
Integral surface	1×2.5 cm			

So, it can be seen from the above tables that there are 6 different antenna combinations and 18 different pairs of tissue and matching layer permittivity and 2 separate matching layer dimensions. Thus, a total of 216 many(=6×18×2) different simulation points are run in HFSS. The simulation points are formed as a table in MATLAB and imported as an optimetric table by HFSS.

Two simulation results are calculated through far-field sweep (FF_Sweep) to compute the delivered average power, and S-parameter sweep (SParam_Sweep) to determine the S11 values at the frequency of 2.4GHz. The simulation for all combinations takes almost 24 hours to run.

4.2 Data Extraction from Simulation Results

For each simulation result, the delivered average power to the integral surface which is calculated through Poynting vector and the S11 value at the frequency of 2.4GHz are extracted and saved to a file. The extracted results are appended to corresponding rows of the table which is formed for the optimetric table via a MATLAB script. The generated dataset is given in the Appendix-B.

5 Gaussian Process Regression Model

In the project proposal, it was mentioned that the machine learning method of an artificial neural network approach will be considered as an estimation model for delivered average powers and S11 values. However, a working model that was constructed with an artificial neural network couldn't be obtained. Therefore, as mentioned in the proposal, alteration of the estimation method to a new approach will be considered if the proposed method fails to provide expected results. So, another machine learning method of regression is chosen as the estimation method.

In a simple regression approach, the estimation of the output variable is achieved by linearly relating the input parameters to the predicted output variable. The weights of the distinct input parameters are calculated as a measurement of how related the output variable is to them. So, these weights are used to estimate the output variable. Throughout the training process, the weights are determined as the ones which give the minimum square root error.

In the estimation model of the S11 values at the frequency of 2.4GHz, the input parameters are selected as circle radius and width, tissue permittivity, matching layer permittivity and width; which the whole data points are shown in the Table 2 and 3. And in the estimation model of delivered average power, the input parameters are the same as the previous one except for additional input as S11 values. So, in the prediction process, first, the S11 values of the corresponding combination are estimated; then with the addition of this value, the delivered average power is estimated.

Since in the project the relationship between input parameters and the output is not linear, the simple linear regression doesn't perform well. Therefore, the estimation model is constructed with a more sophisticated regression approach of Gaussian Process.

5.1 Constructing the Gaussian Process Regression Models

The Gaussian Process Regression model is a probabilistic supervised machine learning approach. The model is able to make estimations through prior knowledge and to give uncertainty measurement of estimations [9]. Namely, the Gaussian Process model is a Gaussian probability distribution over possible functions that can fit the dataset points.

Since the probability distribution over possible functions is known by the model, the means as a function, and the variances to determine how confident the estimations are can be calculated.

In order to construct the estimation models, the Regression Learner App of MATLAB's Statistics and Machine Learning Toolbox [10] is used. Also, 10 K-Fold cross-validation is considered to analyze all the dataset points as test data. Namely, The dataset is divided into 10 distinct parts and each time one part is set as test points and the reaming parts are set as training points; and this training is repeated 10 times to cover all the dataset.

5.1.1 Model to Estimate S11 Values at 2.4GHz

The Regression Learner App tries to optimize the hyperparameters of the Gaussian Process Regression models. The iterations of tried models for estimation of S11 values versus minimum mean square error are shown in the Figure 30. So, it can be seen from the figure that the minimum error is obtained in the 11th iterations where the sigma is 45.6113, the basis function is zero, meaning that there is no basis function in the model, the kernel function is nonisotropic matern 5/2, which defines how the response at one point is affected by responses at other points, and the dataset is standardized through training, meaning that the points values are normalized. The obtained root mean squared error is 0.5917. The estimations of S11 values are shown in the Figure 31.

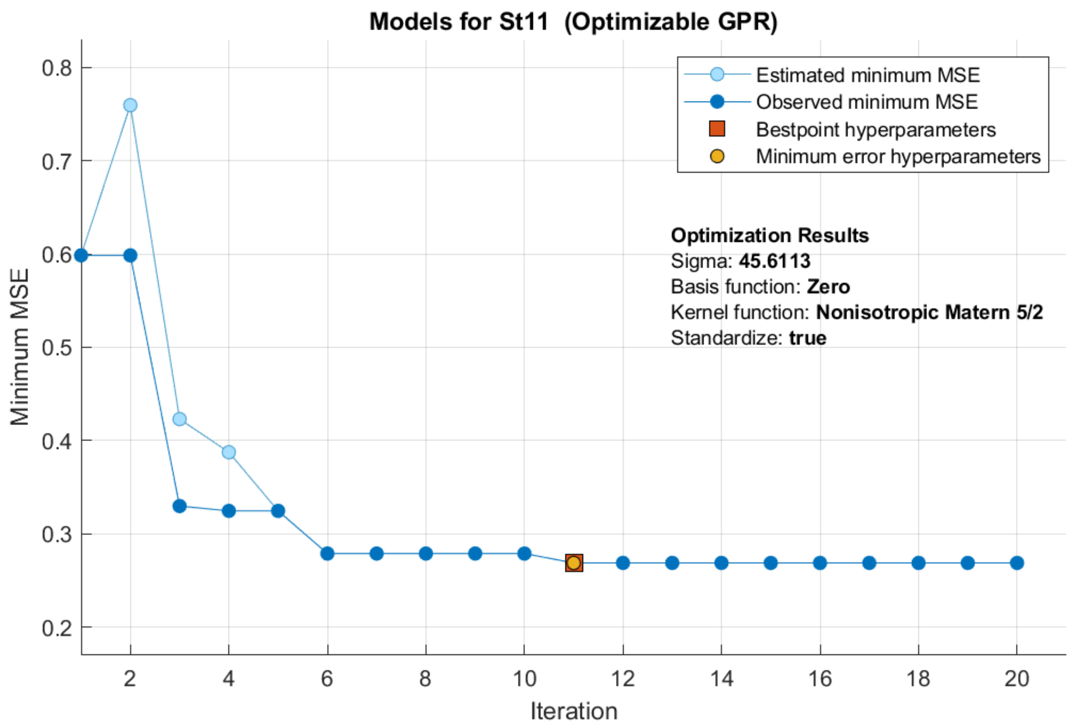


Figure 30: Models for S11 values at 2.4GHz

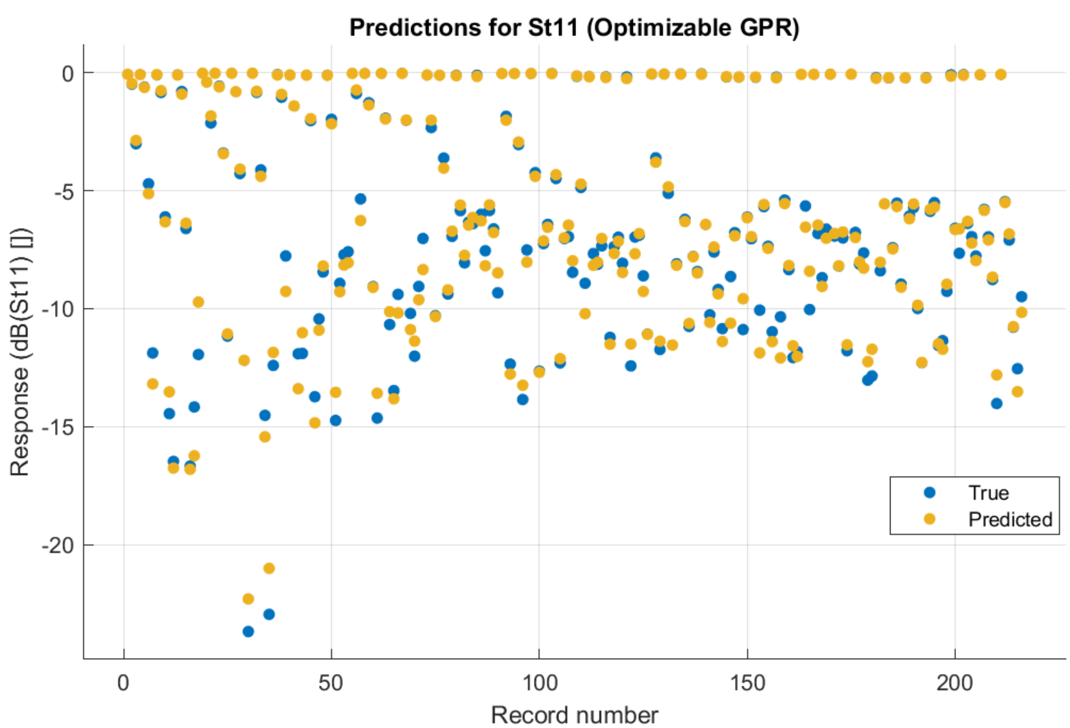


Figure 31: Estimations for S11 values at 2.4GHz

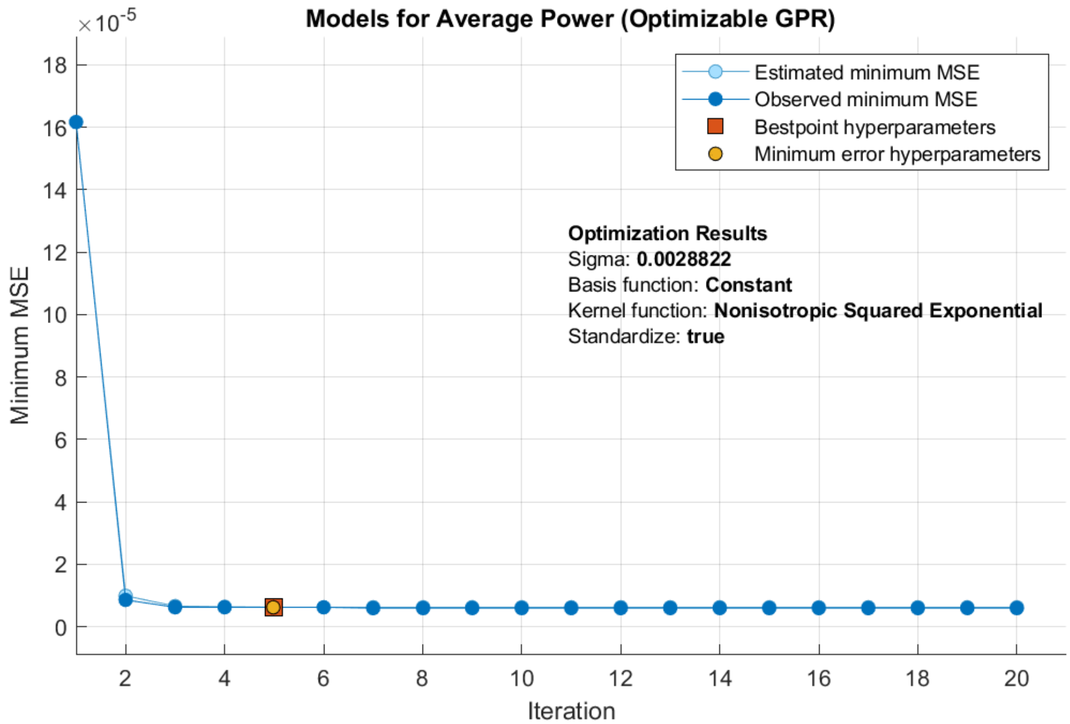


Figure 32: Models for delivered average power

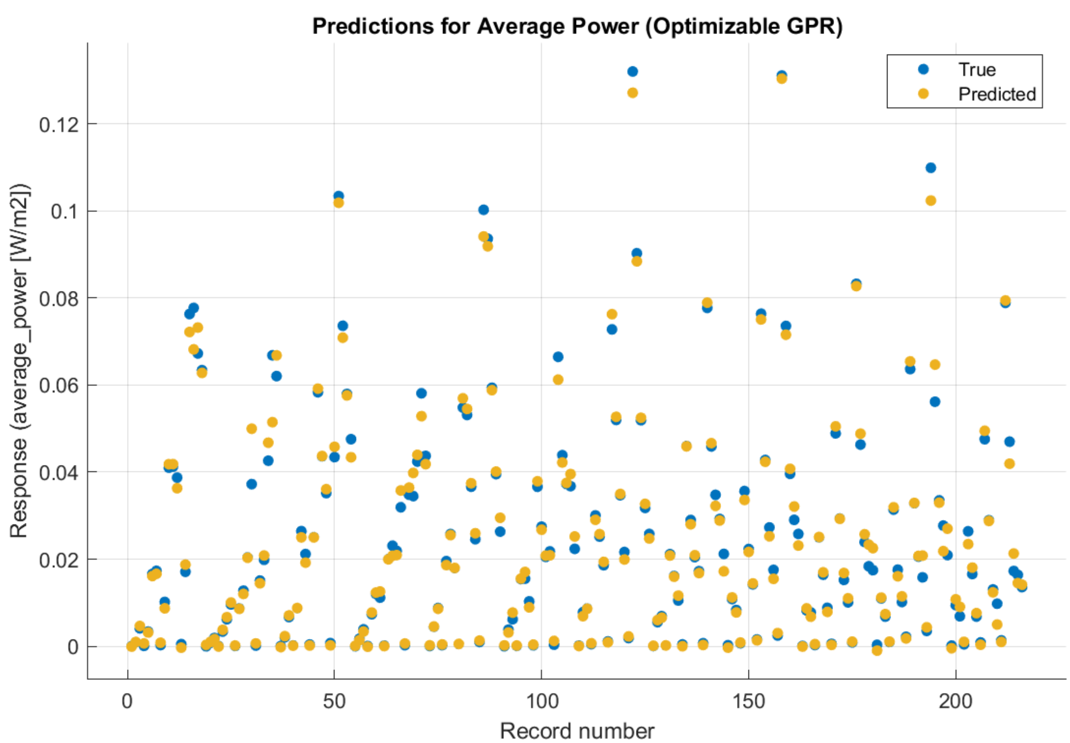


Figure 33: Estimations for delivered average power

5.1.2 Model to Estimate Delivered Average Power

Similarly, the Regression Learner App tries to optimize the hyperparameters of the Gaussian Process Regression models. The iterations of tried models for estimation of delivered average power versus minimum mean square error are shown in the Figure 32. So, it can be seen from the figure that the minimum error is obtained in the 5th iterations where the sigma is 0.00288, the basis function is constant, meaning that the basis function in the model is 1, the kernel function is non-isotropic squared exponential, which defines how the response at one point is affected by responses at other points, and the dataset is standardized through training, meaning that the points values are normalized. The obtained root mean squared error is 0.0025. The estimations of delivered average power values are shown in the Figure 33.

5.2 Determining the Best Parameters in terms of Tissue Permittivity

A MATLAB script is written to determine the best combination of the input parameters given with the tissue permittivity. First, the Gaussian Process Regression models described in the previous sections are constructed. Then, all possible candidate combinations of parameters with given precision and tissue permittivity are generated. After obtaining S11 values for each combination through the first model built in the previous section, the delivered average power is estimated via the second model built in the previous section. Then, the combination which gives the maximum delivered average power is determined. Finally, the script outputs the maximum estimated delivered average power and its input parameters.

The MATLAB scripts which are used in the project are given in the Appendix-C.

6 Results

With the algorithm which is given in the previous section, the best combinations of input parameters are estimated for the tissue permittivity of [30 40 50] and given in the Table 4.

Table 4: Best combinations of parameters obtained from the algorithm

	Values		
Tissue permitivity	30	40	50
Matching layer permitivity	7	8	8
Matching layer width (cm)	0.5	0.5	0.5
circle radius (cm)	1.03	1.00	0.98
circle width (cm)	0.10	0.10	0.10
S11 value (dB)	-9.47	-11.3	-11.9
Average power (W/m2)	0.0365	0.0795	0.136

Also, in order to observe the accuracy of the algorithm, the same input parameters are simulated in HFSS and given in the Table 5.

Table 5: Best combinations of parameters obtained from HFSS simulation results

	Values		
Tissue permitivity	30	40	50
Matching layer permitivity	7	8	8
Matching layer width (cm)	0.5	0.5	0.5
circle radius (cm)	1.03	1.00	0.98
circle width (cm)	0.10	0.10	0.10
S11 value (dB)	-9.92	-9.05	-11.2
Average power (W/m2)	0.0356	0.0767	0.142

It can be observed from the Table 4 and 5 that the estimations of delivered average power which made by the algorithm of Gaussian Process Regression models are very close to the actual simulation results obtained through HFSS.

Furthermore, in order to inspect the effects of placing a matching layer between the antenna and tissue, the same input parameters, except the matching layer permittivity, (it is set to 1 to simulate the layer as free space.) are simulated and given in the Table 6.

Table 6: Best combinations of parameters without the matching layer

	Values		
Tissue permitivity	30	40	50
Matching layer permitivity	1	1	1
Matching layer width (cm)	0.5	0.5	0.5
circle radius (cm)	1.03	1.00	0.98
circle width (cm)	0.10	0.10	0.10
S11 value (dB)	-0.15	-0.19	-0.20
Average power (W/m2)	0.0009	0.0019	0.0037

It can be seen from the Table 6 that without a matching layer, the planar circular loop antenna cannot work properly. Actually, it is not even matched to 50Ω, since S11 values are so close to zero dB. In fact, the matter of unmatched loop antenna in the free space was encountered at the beginning of the project, and the solution of adding a matching network circuit was proposed. Thus, this result is not unexpected.

7 Conclusion

In body-centric wireless communications, the wearable and implantable antennas are highly impacted by the presence of the body due to the high permittivity and lossy characteristics of the human body. So, in order to reduce the effect of human body and to improve the quality of the communication performance, a matching layer is introduced between the antenna and the human body.

The antenna setup is constructed as a circular loop antenna working at a resonance frequency of 2.4GHz. In order to match the antenna to 50Ω, a matching network circuit consisting of a capacitor and/or an inductor can be used. However, the complexity of adding a matching network outweighs the improvement of transmission which is observed by calculating the delivered average power.

With the designed circular loop antenna, a simulation model in HFSS is formed using tissue and matching layers. In order to determine the optimum matching layer parameters and the location of the antenna, the simulation should run for a very long time. So, an approach of machine learning is applied to estimate the best combinations of parameters. The estimation models are constructed by using the Gaussian Process Regression approach. As observed in the result section, the built models can predict very close the simulation results, they reduce the time spent for simulations to obtain the best combination of parameters for the matching layer.

7.1 Realistic Constraints

How many hours a week were required to accomplish this project and the current level of knowledge of mine were the main constraints of the project. I planed to work 16 hours a week but it would lengthen or shorten depending on other things. Another restriction was my knowledge about this field, I needed to spend a couple of weeks reviewing the literature to gain an insight on the machine learning approach to the wearable and implantable antenna field.

7.1.1 Social, Environmental and Economic Impact

The estimation of the effect of matching layers using a machine learning approach can save the time which is spent on determining the parameters of the matching layer using simulation tools. So, the antenna designer will have more time to work on the other aspects of the antenna.

7.1.2 Cost Analysis

There exist some computer tools to realize the project such as a simulation tool, ANSYS HFSS was used and its license is provided by the university; a platform to run the estimation algorithm, MATLAB was used and also its license is provided by the university.

7.1.3 Standards

The project complied with IEEE, IET, EU and Turkish standards and the engineering code of conduct was followed through the project.

A Appendix - Matching Network

A given impedance can be matched to 50Ω with two reactive passive components, which are inductors and capacitors. In order to illustrate the effects of these elements on the input impedance, the Smith Chart is used. There are four possible ways of connecting these components to the load which will be matched [11]:

- *a series inductor* moves the impedance along the constant resistance circle clockwise. The inductor value needed to move the reactance on the Smith chart by a factor of X_L is given by

$$L = \frac{X_L}{2\pi f}$$

- *a series capacitor* moves the impedance along the constant resistance circle in a counter-clockwise direction. The capacitor value needed to move the reactance on the Smith chart by a factor of X_C is given by

$$C = \frac{-1}{2\pi f X_C}$$

- *a shunt inductor* moves the impedance along the constant conductance circle in an counterclockwise direction. The inductor value needed to move the conductance by Y_L is given by

$$L = \frac{-1}{2\pi f Y_L}$$

- *a shunt capacitor* moves the impedance along the constant conductance circle in a clockwise direction. The capacitor value needed to move the conductance by Y_C is given by

$$C = \frac{Y_C}{2\pi f}$$

The Figure 34 shows above mentioned changes on the Smith Chart.

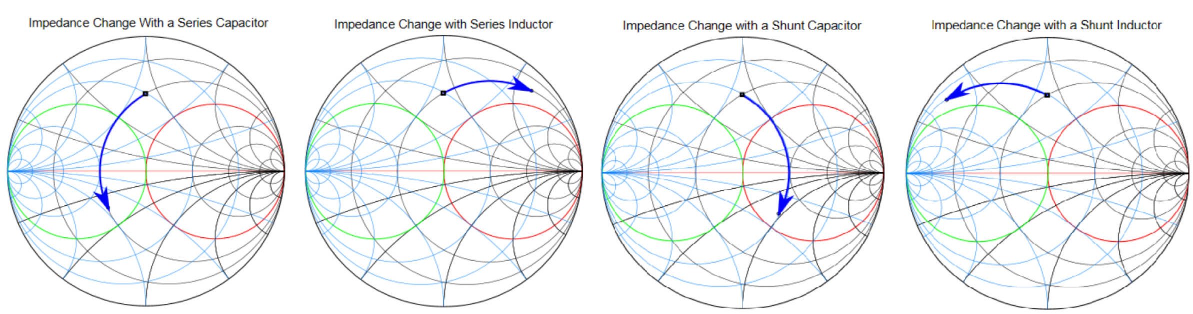


Figure 34: Smith charts depicting impedance changes with the addition of reactance [11]

The first step of the impedance matching is to bring the impedance to the 50Ω (or 20mS) circle; then, the next step is to move the impedance to the 50Ω point (the center of the chart).

B Appendix - Dataset

Table 7: Dataset used in training the estimation model

$muscle_per []	$match_per []	circ_rad [cm]	circ_wid [cm]	match_wid [cm]	average_power [W/m2]	dB(St11) []
20	1	0.6	0.1	0.5	3.85731090787231e-05	-0.0653271641845455
20	7	0.6	0.1	0.5	0.000943869707451686	-0.482921388977102
20	14	0.6	0.1	0.5	0.00412591587141394	-3.00944501651439
30	1	0.6	0.1	0.5	0.000119080026026643	-0.0743365409019561
30	8	0.6	0.1	0.5	0.0033797200126025	-0.605004230568119
30	16	0.6	0.1	0.5	0.016611896220764	-4.70608158820159
30	23	0.6	0.1	0.5	0.0173344433502206	-11.8820393590728
40	1	0.6	0.1	0.5	0.000291212234568926	-0.0877545448928181
40	9	0.6	0.1	0.5	0.0101836684076385	-0.835860199737677
40	17	0.6	0.1	0.5	0.041020609584019	-6.107891554945
40	24	0.6	0.1	0.5	0.0412637289700834	-14.4535321605086
40	32	0.6	0.1	0.5	0.0387337004087578	-16.4764538090978
50	1	0.6	0.1	0.5	0.000508998747823733	-0.0898710617390946
50	9	0.6	0.1	0.5	0.0171102007877218	-0.794900713574893
50	17	0.6	0.1	0.5	0.0762906983765163	-6.60400875419262
50	26	0.6	0.1	0.5	0.0776935145494212	-16.6772290443383
50	34	0.6	0.1	0.5	0.0672409662766033	-14.1691004694149
50	42	0.6	0.1	0.5	0.0634051049439861	-11.9450307666366
20	1	0.6	0.1	1	1.95163947385096e-05	-0.0200945838288544
20	7	0.6	0.1	1	0.000810638939940769	-0.39718885079676
20	14	0.6	0.1	1	0.00192311560554634	-2.12848580750207
30	1	0.6	0.1	1	5.33070256894552e-05	-0.0203987878556685
30	8	0.6	0.1	1	0.00340050540705951	-0.570993070560003
30	16	0.6	0.1	1	0.00621595297385382	-3.40197774892864
30	23	0.6	0.1	1	0.00962169890871908	-11.1662350181376
40	1	0.6	0.1	1	0.000108224051964707	-0.0197376233701562
40	9	0.6	0.1	1	0.00869073507483627	-0.803133906229158
40	17	0.6	0.1	1	0.0127946344804175	-4.27341162518322
40	24	0.6	0.1	1	0.0204130341723874	-12.1966545827068
40	32	0.6	0.1	1	0.0372400129022707	-23.6889272567115
50	1	0.6	0.1	1	0.000179671629240572	-0.0189887045635462
50	9	0.6	0.1	1	0.0151007820517465	-0.831735593570685
50	17	0.6	0.1	1	0.0198754518204071	-4.11610447123976

50	26	0.6	0.1	1	0.0426213689421806	-14.5204582656281
50	34	0.6	0.1	1	0.0668168872041315	-22.9602579917891
50	42	0.6	0.1	1	0.0620397647700101	-12.4068275292584
20	1	0.7	0.1	0.5	6.53197356171225e-05	-0.0789307352395015
20	7	0.7	0.1	0.5	0.00209476663639536	-1.03894969422301
20	14	0.7	0.1	0.5	0.00672051195903444	-7.76557897434602
30	1	0.7	0.1	0.5	0.000207494062653162	-0.0959581908183526
30	8	0.7	0.1	0.5	0.00878306859632577	-1.410416509168
30	16	0.7	0.1	0.5	0.0264542920072808	-11.9160283213103
30	23	0.7	0.1	0.5	0.021194150358583	-11.8996328400016
40	1	0.7	0.1	0.5	0.000433314245740518	-0.102547667568958
40	9	0.7	0.1	0.5	0.0250552359062866	-2.02947812624896
40	17	0.7	0.1	0.5	0.0583338443377532	-13.7342171680242
40	24	0.7	0.1	0.5	0.0436375989263163	-10.434068524087
40	32	0.7	0.1	0.5	0.0351795122080166	-8.44507687440366
50	1	0.7	0.1	0.5	0.000742811955031755	-0.103297444518867
50	9	0.7	0.1	0.5	0.0434354200757833	-1.97189738304512
50	17	0.7	0.1	0.5	0.103370607597333	-14.7393197518239
50	26	0.7	0.1	0.5	0.0735953880897122	-8.9276080675225
50	34	0.7	0.1	0.5	0.0579511484053635	-7.71504249473308
50	42	0.7	0.1	0.5	0.0475637580980338	-7.58700375422141
20	1	0.7	0.1	1	2.94414171604847e-05	-0.0292924577268053
20	7	0.7	0.1	1	0.00174865997413713	-0.880123096599795
20	14	0.7	0.1	1	0.00389935756176448	-5.35046519613288
30	1	0.7	0.1	1	8.16447472037369e-05	-0.0281536251912577
30	8	0.7	0.1	1	0.00738228762465462	-1.27218674654265
30	16	0.7	0.1	1	0.0120567487265493	-9.06834934223318
30	23	0.7	0.1	1	0.011177953207653	-14.6393027986728
40	1	0.7	0.1	1	0.000168716454598996	-0.0272314212041773
40	9	0.7	0.1	1	0.0200057953183912	-1.92617500441717
40	17	0.7	0.1	1	0.0230985943392814	-10.6744257033522
40	24	0.7	0.1	1	0.0219069134584672	-13.4737756784978
40	32	0.7	0.1	1	0.0319315480764343	-9.39311216622137
50	1	0.7	0.1	1	0.000270466354584622	-0.025357688358214
50	9	0.7	0.1	1	0.0348116944681649	-2.01527839857656
50	17	0.7	0.1	1	0.0344841371887724	-10.2033199539699
50	26	0.7	0.1	1	0.0424237268925241	-12.0147978561197
50	34	0.7	0.1	1	0.0581008588004581	-9.05945067949067
50	42	0.7	0.1	1	0.0437381715879287	-7.02573258412863
20	1	0.8	0.1	0.5	9.2721910232044e-05	-0.0978248467602957
20	7	0.8	0.1	0.5	0.00449632396009254	-2.32408059097992
20	14	0.8	0.1	0.5	0.00878875922161244	-10.30101226235

30	1	0.8	0.1	0.5	0.000271399913396062	-0.104500916946848
30	8	0.8	0.1	0.5	0.0195605365123188	-3.6164147844143
30	16	0.8	0.1	0.5	0.025793222704863	-9.38058283340277
30	23	0.8	0.1	0.5	0.0179772897417327	-6.93843439576752
40	1	0.8	0.1	0.5	0.000550898409748944	-0.108008877237553
40	9	0.8	0.1	0.5	0.0548201587042969	-5.84593843891462
40	17	0.8	0.1	0.5	0.0531244134789074	-8.0579430036812
40	24	0.8	0.1	0.5	0.0366367090601372	-6.33729425731865
40	32	0.8	0.1	0.5	0.0246037692120119	-6.39697138613259
50	1	0.8	0.1	0.5	0.00100835232070393	-0.108373407469182
50	9	0.8	0.1	0.5	0.100220886652384	-5.9992630559491
50	17	0.8	0.1	0.5	0.0935825292344857	-7.54578017365479
50	26	0.8	0.1	0.5	0.0593682346073535	-5.83873834194741
50	34	0.8	0.1	0.5	0.039533765751045	-6.61627741573299
50	42	0.8	0.1	0.5	0.0263337213646664	-9.33074244775781
20	1	0.8	0.1	1	3.95711917211941e-05	-0.0325124524799622
20	7	0.8	0.1	1	0.00383243021127423	-1.84424074675478
20	14	0.8	0.1	1	0.0062076240517612	-12.3506597780411
30	1	0.8	0.1	1	0.000119428344984	-0.0318424316058091
30	8	0.8	0.1	1	0.0154539015533835	-3.04582885042448
30	16	0.8	0.1	1	0.0155359001358465	-13.8519566454979
30	23	0.8	0.1	1	0.0103251070105866	-7.51117205162813
40	1	0.8	0.1	1	0.000249370913030741	-0.032682002580466
40	9	0.8	0.1	1	0.0366386126120918	-4.23374421453009
40	17	0.8	0.1	1	0.027485275069014	-12.6485915915262
40	24	0.8	0.1	1	0.0205444362466268	-7.24207220842788
40	32	0.8	0.1	1	0.0217961428664977	-6.42734733166321
50	1	0.8	0.1	1	0.000385506475216273	-0.0303014926194012
50	9	0.8	0.1	1	0.0664598058641094	-4.4843809082271
50	17	0.8	0.1	1	0.0438986165858303	-12.3002217241887
50	26	0.8	0.1	1	0.0372763806356604	-7.03038462888823
50	34	0.8	0.1	1	0.0367709876991446	-6.95004661898651
50	42	0.8	0.1	1	0.0223786802048946	-8.46233047308901
20	1	0.9	0.2	0.5	0.00017676253081061	-0.151321964427438
20	7	0.9	0.2	0.5	0.00784554545893351	-4.86008347427319
20	14	0.9	0.2	0.5	0.00867212413427703	-8.92444365854232
30	1	0.9	0.2	0.5	0.000507176025749906	-0.162520438722039
30	8	0.9	0.2	0.5	0.0300656952635097	-7.66549364701983
30	16	0.9	0.2	0.5	0.0251980180383246	-8.12003360840763
30	23	0.9	0.2	0.5	0.0186305533148099	-7.33962530516815
40	1	0.9	0.2	0.5	0.001139843842424	-0.178219558646462
40	9	0.9	0.2	0.5	0.0727734965674622	-11.2161568973314

40	17	0.9	0.2	0.5	0.0519838268080056	-7.36890227185662
40	24	0.9	0.2	0.5	0.0347025038856724	-6.96997380627694
40	32	0.9	0.2	0.5	0.0216626344672313	-8.0730636197749
50	1	0.9	0.2	0.5	0.00191465069821657	-0.178776999544922
50	9	0.9	0.2	0.5	0.131978373738287	-12.4283086682852
50	17	0.9	0.2	0.5	0.090242555030077	-6.96274266094226
50	26	0.9	0.2	0.5	0.051904986217589	-6.89074343746513
50	34	0.9	0.2	0.5	0.0317712578877142	-8.61133647904926
50	42	0.9	0.2	0.5	0.0258037910589913	-11.0782695721352
20	1	0.9	0.2	1	8.39555827874642e-05	-0.0567604774352848
20	7	0.9	0.2	1	0.00555277296915799	-3.60701723019477
20	14	0.9	0.2	1	0.0069316621326699	-11.7339748332303
30	1	0.9	0.2	1	0.000207423740958727	-0.0545119756699984
30	8	0.9	0.2	1	0.0211704939234634	-5.1107966580568
30	16	0.9	0.2	1	0.0161544059418048	-11.543413818034
30	23	0.9	0.2	1	0.010529129716158	-8.10027509577762
40	1	0.9	0.2	1	0.00044776977344767	-0.0514133109375809
40	9	0.9	0.2	1	0.0459834458218958	-6.21734855315302
40	17	0.9	0.2	1	0.0290011465447741	-10.7560777309637
40	24	0.9	0.2	1	0.0204881517870948	-7.78518591378522
40	32	0.9	0.2	1	0.0172701610183353	-8.42617498704316
50	1	0.9	0.2	1	0.000715432274461757	-0.0496187752377736
50	9	0.9	0.2	1	0.0777110975740773	-6.43232372412858
50	17	0.9	0.2	1	0.04590000521723	-10.2627687068918
50	26	0.9	0.2	1	0.0347888882604916	-7.59744319673099
50	34	0.9	0.2	1	0.0292427106559155	-9.18780940091298
50	42	0.9	0.2	1	0.0212212918297339	-10.8490644703293
20	1	1	0.2	0.5	0.000260571507512578	-0.176994388456258
20	7	1	0.2	0.5	0.0108518483828812	-8.63869761080799
20	14	1	0.2	0.5	0.00833680958500589	-6.77733922546122
30	1	1	0.2	0.5	0.000729839289656775	-0.185370821297338
30	8	1	0.2	0.5	0.0356445911087652	-10.886900521397
30	16	1	0.2	0.5	0.0223368034930663	-6.11739861506029
30	23	1	0.2	0.5	0.0142345712640643	-7.03561087951035
40	1	1	0.2	0.5	0.00158588180117511	-0.190421666982219
40	9	1	0.2	0.5	0.0763597589576843	-10.0640434648319
40	17	1	0.2	0.5	0.0427798823629048	-5.67482485208827
40	24	1	0.2	0.5	0.0273023482954024	-7.35551169433632
40	32	1	0.2	0.5	0.0175589429580838	-10.9819696774375
50	1	1	0.2	0.5	0.00249426828174967	-0.187234685406654
50	9	1	0.2	0.5	0.13104998236474	-10.3484107499745
50	17	1	0.2	0.5	0.0735606738853006	-5.39541325003458

50	26	1	0.2	0.5	0.0396059121152642	-8.34488452218731
50	34	1	0.2	0.5	0.0290378377749525	-12.076902083775
50	42	1	0.2	0.5	0.0258005322405383	-11.8186519172443
20	1	1	0.2	1	9.83041541706646e-05	-0.0688925477094852
20	7	1	0.2	1	0.00822901253592333	-5.64865258281494
20	14	1	0.2	1	0.00777714297666822	-10.0386661174457
30	1	1	0.2	1	0.000304126190791683	-0.0691938001516228
30	8	1	0.2	1	0.0250369496692469	-6.81651983364555
30	16	1	0.2	1	0.0164451940563156	-8.68640599411947
30	23	1	0.2	1	0.0088436223711379	-6.63079930377768
40	1	1	0.2	1	0.000579401899959716	-0.0640173956572352
40	9	1	0.2	1	0.0489051895428289	-6.9247187847918
40	17	1	0.2	1	0.0293337517455858	-8.19779254612981
40	24	1	0.2	1	0.0152475657051358	-7.00261260786436
40	32	1	0.2	1	0.0100967186737171	-11.7844111240203
50	1	1	0.2	1	0.000900410900417935	-0.0610173307331685
50	9	1	0.2	1	0.0832273919631568	-6.76422635157617
50	17	1	0.2	1	0.0463157451920356	-8.11387251909546
50	26	1	0.2	1	0.0239394799453624	-7.63817485127724
50	34	1	0.2	1	0.0183873753233987	-13.0320973136396
50	42	1	0.2	1	0.0174818258093266	-12.8568963888988
20	1	1.1	0.2	0.5	0.000349363003136874	-0.203216020765639
20	7	1.1	0.2	0.5	0.0110133262719112	-8.39635686226305
20	14	1.1	0.2	0.5	0.00683421208462141	-5.5559201346298
30	1	1.1	0.2	0.5	0.00100262480167156	-0.214782972600381
30	8	1.1	0.2	0.5	0.031392038582835	-7.416966435166
30	16	1.1	0.2	0.5	0.0176035094295097	-5.52532665728657
30	23	1.1	0.2	0.5	0.0101703391279924	-8.95932073605815
40	1	1.1	0.2	0.5	0.0021937910988412	-0.214926814327446
40	9	1.1	0.2	0.5	0.0636562664369678	-6.07503848273599
40	17	1.1	0.2	0.5	0.0328754109092496	-5.70991072366207
40	24	1.1	0.2	0.5	0.0206105112582009	-9.99443521674554
40	32	1.1	0.2	0.5	0.0158479510057365	-12.2901001884554
50	1	1.1	0.2	0.5	0.00354513028190085	-0.20959827607812
50	9	1.1	0.2	0.5	0.109859846816615	-5.8730701585435
50	17	1.1	0.2	0.5	0.0561606686188603	-5.50649643761209
50	26	1.1	0.2	0.5	0.0335149926520659	-11.5630335656693
50	34	1.1	0.2	0.5	0.0277230588515506	-11.3564501129782
50	42	1.1	0.2	0.5	0.0209441489148876	-9.26886180401483
20	1	1.1	0.2	1	0.000164236882660907	-0.0888272306304959
20	7	1.1	0.2	1	0.00946834117121069	-6.59463457177887
20	14	1.1	0.2	1	0.00690410819947819	-7.65226367464534

30	1	1.1	0.2	1	0.000469071386566768	-0.0845303531255714
30	8	1.1	0.2	1	0.0264199712743261	-6.38605234784711
30	16	1.1	0.2	1	0.0165975091103893	-6.9477023915914
30	23	1.1	0.2	1	0.0068405776237264	-7.76758164784336
40	1	1.1	0.2	1	0.000899004820799243	-0.0813001603495266
40	9	1.1	0.2	1	0.0475519401025238	-5.7925742384324
40	17	1.1	0.2	1	0.0289549072323964	-6.95805337192994
40	24	1.1	0.2	1	0.0130681462343176	-8.76567573614954
40	32	1.1	0.2	1	0.00979293131407492	-14.0235226130341
50	1	1.1	0.2	1	0.00136069224006804	-0.0758355000600765
50	9	1.1	0.2	1	0.0788583133270711	-5.46487742939609
50	17	1.1	0.2	1	0.0469930439161307	-7.08850841359112
50	26	1.1	0.2	1	0.0173031096749947	-10.7873040788972
50	34	1.1	0.2	1	0.0164153828420974	-12.5475178207727
50	42	1.1	0.2	1	0.0135967147043387	-9.49318854722603

C Appendix - MATLAB Scripts

The MATLAB script which is used to determine the dataset points and to generate the dataset is given in the following:

```
%% Forming the dataset point for simulation in HFSS

dataset=cell(218,11);
dataset{1,1}="$muscle_per";
dataset{1,2}="$match_per";
dataset{1,3}="circ_rad";
dataset{1,4}="circ_wid";
dataset{1,5}="port_gap";
dataset{1,6}="turn_deg";
dataset{1,7}="turn_z";
dataset{1,8}="muscle_dim";
dataset{1,9}="match_dim";
dataset{1,10}="match_wid";
dataset{1,11}="surface_dim";
dataset{2,1}="";
dataset{2,2}="";
dataset(2,3:end)={"cm"};
dataset{2,6}="rad";
dataset{2,7}="deg";

circ_rad = [0.6 0.7 0.8 0.9 1 1.1];
circ_wid = [0.1 0.1 0.1 0.2 0.2 0.2];
port_gap = 0.1;
turn_deg = (2*pi*circ_rad)./(2*pi*circ_rad+port_gap)*2*pi;
turn_z = 0;
muscle_dim = 4;
match_dim = 5;
match_wid = [0.5 1];
surface_dim = 1;

mu_p = [20 30 40 50];
ma_p = [1 7 14 0 0 0 ; 1 8 16 23 0 0 ; 1 9 17 24 32 0; 1 9 17 26 34 42];

ind=3;
for i=1:6
    for j=1:2
        for k=1:4
            for l = 1:6
                if ma_p(k,l)==0
                    break;
                end
                dataset(ind,1)={mu_p(k)};
                dataset(ind,2)={ma_p(k,l)};
                dataset(ind,3)={circ_rad(i)};
                dataset(ind,4)={circ_wid(i)};
                dataset(ind,5)={port_gap};
```

```
                dataset(ind,6)={turn_deg(i)};
                dataset(ind,7)={turn_z};
                dataset(ind,8)={muscle_dim};
                dataset(ind,9)={match_dim};
                dataset(ind,10)={match_wid(j)};
                dataset(ind,11)={surface_dim};
                ind=ind+1;
            end
        end
    end
end

%% Arranging the dataset for further processing

dataset(:,[5 6 7 8 9 11])=[];
dataset(2,:)=[];
dataset{1,1}="$muscle_per []";
dataset{1,2}="$match_per []";
dataset{1,3}="circ_rad [cm]";
dataset{1,4}="circ_wid [cm]";
dataset{1,5}="match_wid [cm]";

% Adding corresponding average power calculation values
dataset{1,6}="average_power [W/m2]";
dataset(2:end,6)=num2cell(load("AveragePower.txt"));
%%
% Adding corresponing S11 values at 2.4GHz in dB

dataset{1,7}="dB(St11) []";
returnLoss = readmatrix("Return Loss.csv");
ind=2;
for i=1:6
    for j=1:2
        for k=1:4
            for l = 1:6
                if ma_p(k,l)==0
                    break;
                end
                indx=find(ismember(returnLoss(:,1:5), [ma_p(k,l) ...
                    mu_p(k) circ_rad(i) circ_wid(i) match_wid(j)],'rows'));
                dataset{ind,7}=returnLoss(indx,7);
                ind=ind+1;
            end
        end
    end
end

%writecell(dataset,'data.csv');
%save("data.mat", "dataset");
```

The MATLAB script which is used to determine the best combination of the input parameters given with the tissue permittivity is given in the following:

```
file_path = fullfile(pwd(),'data.csv');
data_table = readtable(file_path,'PreserveVariableNames', true);

muscle_per = 50;
percision = 0.01;

[trainedModel_St11, validationRMSE_St11] = trainRegressionModelSt11(data_table);

[trainedModel_AP, validationRMSE_AP] = trainRegressionModelAP(data_table);


candidate_combinations = gen_candidate(muscle_per,percision,trainedModel_St11);

estimate_AP = trainedModel_AP.predictFcn(candidate_combinations);

best = candidate_combinations(estimate_AP==max(estimate_AP),:);

fprintf("The maximum delivered average power for the muscle permittivity" ...
  + " of %i is estimated as %f with the following combination of inputs: \n",...
  muscle_per, max(estimate_AP));
disp(best);
```

The MATLAB function which is used to form all possible combinations of input parameters given with tissue permittivity, precision and the model to estimate the S11 values is given in the following:

```
function [candidate] = gen_candidate(muscle_per,precision,trainedModel_St11)
% [candidate] = gen_candidate(muscle_per,precision,trainedModel_St11)
% returns the all possible candidate combinations of inputs with specified
% muscle permittivity and desired precision of the inputs.
%
% Since the St11 values are also an output of the simulation results,
% a trained model should be provided for estimation of the St11 values.

match_per = 1:muscle_per;
circ_rad = 0.6:precision:1.1;
circ_wid = 0.1:precision:0.2;
match_wid = 0.5:0.1:1.0;

candidate = array2table(combvec(muscle_per,match_per,circ_rad,circ_wid,match_wid)');
candidate.Properties.VariableNames = [{'$muscle_per []'} {'$match_per []'} ...
    {'circ_rad [cm]'}  {'circ_wid [cm]'}  {'match_wid [cm]'} ];

candidate.('dB(St11) []') = trainedModel_St11.predictFcn(candidate);

end
```

The MATLAB function which is used to construct the Gaussian Process Regression model to estimate the S11 values at the frequency of 2.4GHz is given in the following:

```
function [trainedModel, validationRMSE] = trainRegressionModelSt11(trainingData)
% [trainedModel, validationRMSE] = trainRegressionModelSt11(trainingData)
% returns a trained regression model and its RMSE. This code recreates the
% model trained in Regression Learner app. Use the generated code to
% automate training the same model with new data.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedModel: a struct containing the trained regression model. The
%       struct contains various fields with information about the trained
%       model.
%
%      trainedModel.predictFcn: a function to make predictions on new data.
%
%      validationRMSE: a double containing the RMSE. In the app, the
%       History list displays the RMSE for each model.
%
% Use the code to train the model with new data. To retrain your model,
% call the function from the command line with your original data or new
% data as the input argument trainingData.
%
% For example, to retrain a regression model trained with the original data
% set T, enter:
%   [trainedModel, validationRMSE] = trainRegressionModelSt11(T)
%
% To make predictions with the returned 'trainedModel' on new data T2, use
%   yfit = trainedModel.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'$muscle_per []', '$match_per []', 'circ_rad [cm]', 'circ_wid [cm]',
    'match_wid [cm]'};
predictors = inputTable(:, predictorNames);
response = inputTable.('dB(St11) []');
isCategoricalPredictor = [false, false, false, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
```

```matlab
regressionGP = fitrgp(...
    predictors, ...
    response, ...
    'BasisFunction', 'none', ...
    'KernelFunction', 'ardmatern52', ...
    'Sigma', 45.61126539670443, ...
    'Standardize', true);

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
gpPredictFcn = @(x) predict(regressionGP, x);
trainedModel.predictFcn = @(x) gpPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'$match_per []', '$muscle_per []', 'circ_rad [cm]', 'circ_wid [cm]', 'match_wid [cm]'};
trainedModel.RegressionGP = regressionGP;
trainedModel.About = 'This struct is a trained model exported from Regression Learner R2019b.';
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n yfit = c.predictFcn(T) \nreplacing ''c'' with the name of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table, T, must contain the variables returned by: \n c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype) must match the original training data. \nAdditional variables are ignored. \n \nFor more information, see <a href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''), ''appregression_exportmodeltoworkspace'')">How to predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'$muscle_per []', '$match_per []', 'circ_rad [cm]', 'circ_wid [cm]', 'match_wid [cm]'};
predictors = inputTable(:, predictorNames);
response = inputTable.('dB(St11) []');
isCategoricalPredictor = [false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.RegressionGP, 'KFold', 10);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));
```

The MATLAB function which is used to construct the Gaussian Process Regression model to estimate the delivered average power is given in the following:

```
function [trainedModel, validationRMSE] = trainRegressionModelAP(trainingData)
% [trainedModel, validationRMSE] = trainRegressionModelAP(trainingData)
% returns a trained regression model and its RMSE. This code recreates the
% model trained in Regression Learner app. Use the generated code to
% automate training the same model with new data.
%
%  Input:
%      trainingData: a table containing the same predictor and response
%       columns as imported into the app.
%
%  Output:
%      trainedModel: a struct containing the trained regression model. The
%       struct contains various fields with information about the trained
%       model.
%
%      trainedModel.predictFcn: a function to make predictions on new data.
%
%      validationRMSE: a double containing the RMSE. In the app, the
%       History list displays the RMSE for each model.
%
% Use the code to train the model with new data. To retrain your model,
% call the function from the command line with your original data or new
% data as the input argument trainingData.
%
% For example, to retrain a regression model trained with the original data
% set T, enter:
%   [trainedModel, validationRMSE] = trainRegressionModelAP(T)
%
% To make predictions with the returned 'trainedModel' on new data T2, use
%   yfit = trainedModel.predictFcn(T2)
%
% T2 must be a table containing at least the same predictor columns as used
% during training. For details, enter:
%   trainedModel.HowToPredict

% Auto-generated by MATLAB


% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'$muscle_per []', '$match_per []', 'circ_rad [cm]', 'circ_wid [cm]',
    'match_wid [cm]', 'dB(St11) []'};
predictors = inputTable(:, predictorNames);
response = inputTable.('average_power [W/m2]');
isCategoricalPredictor = [false, false, false, false, false, false];

% Train a regression model
```

```
% This code specifies all the model options and trains the model.
regressionGP = fitrgp(...
    predictors, ...
    response, ...
    'BasisFunction', 'constant', ...
    'KernelFunction', 'ardsquaredexponential', ...
    'Sigma', 0.00288215422761964, ...
    'Standardize', true);

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
gpPredictFcn = @(x) predict(regressionGP, x);
trainedModel.predictFcn = @(x) gpPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'$match_per []', '$muscle_per []', 'circ_rad [cm]', 'circ_wid [cm]', 'dB(St11) []', 'match_wid [cm]'};
trainedModel.RegressionGP = regressionGP;
trainedModel.About = 'This struct is a trained model exported from Regression Learner R2019b.';
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n yfit = c.predictFcn(T) \nreplacing ''c'' with the name of the variable that is this struct, e.g. ''trainedModel''. \n \nThe table, T, must contain the variables returned by: \n c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype) must match the original training data. \nAdditional variables are ignored. \n \nFor more information, see <a href="matlab:helpview(fullfile(docroot, ''stats'', ''stats.map''), ''appregression_exportmodeltoworkspace'')">How to predict using an exported model</a>.');

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'$muscle_per []', '$match_per []', 'circ_rad [cm]', 'circ_wid [cm]', 'match_wid [cm]', 'dB(St11) []'};
predictors = inputTable(:, predictorNames);
response = inputTable.('average_power [W/m2]');
isCategoricalPredictor = [false, false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.RegressionGP, 'KFold', 10);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));
```

Bibliography

[1] Ito K., Lin CH., Lin HY. (2015) Evaluation of Wearable and Implantable Antennas with Human Phantoms. In: Chen Z. (eds) Handbook of Antenna Technologies. Springer, Singapore.

[2] Gabriel, C. (1996). Compilation of the Dielectric Properties of Body Tissues at RF and Microwave Frequencies. http://niremf.ifac.cnr.it/docs/DIELECTRIC/Report.html [Accessed 2 November 2020]

[3] C. A. Balanis, Antenna Theory: Analysis and Design, Chichester, West Sussex:Wiley, 5th chapter, 2016

[4] R. R. Singh, S. P. Diwakar and S. Shastri, "Design of circular loop antenna with step change in loop width," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 2016, pp. 1660-1663, doi: 10.1109/SCOPES.2016.7955724

[5] J. Annovasho, O. Buka, M. Khoiro and Y. H. Pramono, "Design and optimization high-performance bi-circular loop antenna with plane reflector and coaxial feed line at 2.45 GHz frequency," 2017 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), Surabaya, 2017, pp. 154-158, doi: 10.1109/ISSIMM.2017.8124282

[6] P. Deo, A. Mehta, D. Mirshekar-Syahka, P. J. Massey and H. Nakano, "Circular loop antenna for pattern steerable applications," 2009 IEEE Antennas and Propagation Society International Symposium, Charleston, SC, 2009, pp. 1-4, doi: 10.1109/APS.2009.5171565

[7] R. Hasse, W. Hunsicker, K. Naishadham, A. Z. Elseherbeni and D. Kajfez, "Analysis and design of a partitioned circular loop antenna for omni-directional radiation," 2011 IEEE International Symposium on Antennas and Propagation (APSURSI), Spokane, WA, 2011, pp. 1379-1382, doi: 10.1109/APS.2011.5996548

[8] Retrieved 15 December 2020, from https://www.will-kelsey.com/smith_chart/

[9] C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X.

[10] Retrieved 25 January 2021, from https://www.mathworks.com/help/stats/regression-learner-app.html

[11] Retrieved 14 December 2020, from https://www.cypress.com/file/136236/download