**PROJECT REPORT**

***On***

**"DESIGN OF MICROSTRIP ANTENNA USING ARTIFICIAL NEURAL NETWORK MODEL"**

|  |  |
| --- | --- |
|  | *submitted by* |
|  |  |
| **Name** | **SAP No** |
| **1) SAURABH LABDE** | **60002130058** |
| **2) STUTI PATEL** | **60002130080** |
| **3) MEGH SHUKLA** | **60002130114** |
|  |  |

*under the supervision of*

Dr. Amit A Deshmukh

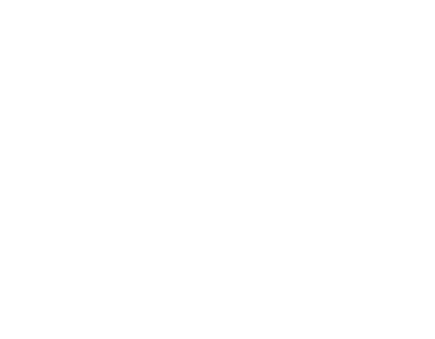
Professor and Head,

Department of Electronics and Telecommunication Engineering

**DEPARTMENT OF**

**ELECTRONICS AND TELECOMMUNICATION ENGINEERING**

**Academic Year: 2016-2017**



**Dwarkadas J. Sanghvi College of Engineering**

**Academic Year: 2016-2017**

**TABLE OF CONTENTS**

DECLARATION…………………………………………... i

CERTIFICATE…………………………………………….. ii

ACKNOWLEDGEMENTS………………………………... iii

ABSTRACT……………………………………………….. iv

LIST OF TABLES………………………………………… v

LIST OF FIGURES………………………………………... vi

INTRODUCTION…………………………………………. 1

* 1. Problem Statement……………………………... 2
  2. Requirement of ANN…………………………... 2

MICROSTRIP ANTENNA……………………………….. 3

2.1 Introduction to MSA…………………………… 3

2.2 Circular Patch…………………………………... 4

2.3 Equilateral Triangle Patch……………………… 6

2.4 Fringing Fields in MSA………………………… 8

ARTIFICIAL NEURAL NETWORKS…………………… 10

3.1 Introduction to ANN……………………………. 10

3.2 Types of ANN…………………………………... 11

3.3 Learning Mechanisms………………………….. 13

3.4 Multilayer Perceptron…………………………... 15

3.5 Backpropagation Learning……………………. 16

IE3D………………………………………………………… 18

4.1 Introduction……………………………………... 18

4.2 Simulation using IE3D………………………….. 19

NEURAL NETWORK PROGRAMMING………………... 20

5.1 Python Programming Language………………… 20

5.2 Numpy Library………………………………….. 21

NEURAL NETWORK ARCHITECTURE………………… 23

6.1 Network parameters…………………………….. 23

6.2 Network I/O……………………………………… 23

6.3 Configuration of Hidden Layer(s)………………. 24

6.4 Training………………………………………….. 25

6.5 Testing Neural Network Data…………………… 26

CONCLUSION……………………………………………… 29

APPENDIX A……………………………………………….. A-1

REFERENCES

**DECLARATION**

*We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature*:

*Name*:

*SAP ID*:

*Date*:

i

**CERTIFICATE**

This is to certify that the Project Stage-II

“DESIGN OF MICROSTRIP ANTENNA USING ARTFICIAL NEURAL NETWORK MODEL”

|  |  |
| --- | --- |
| Submitted By: |  |
| 1) SAURABH LABDE | 60002130058 |
| 2) STUTI PATEL | 60002130080 |
| 3) MEGH SHUKLA | 60002130114 |

Students of Electronics and Telecommunication Engineering have successfully completed their **Project Stage- II** required for the fulfillment of **SEM VIII** as per the norms prescribed by the **University of Mumbai** during the First half of the year 2017. The project synopsis report has been assessed and found to be satisfactory.

**(Internal Guide)** **(External Guide)**

**(Head of Department)** **(Principal)**

**(Internal Examiner)** **(External Examiner)**

ii

**ACKNOWLEDGEMENT**

We would like to thank our Guide, *Dr Amit A Deshmukh* for giving valuable insights into the vast field of Microstrip Antennas. It is his sheer dedication and passion towards Antennas that drove us to complete the project successfully in the stipulated time frame. We are fortunate to study Antennas under a personality renowned in his field.

A project is not complete without a mentor who we could approach for doubts on regular basis. Hence we would like to thank *Prof Venkata Chavali* for contributing her time in clearing our concepts and also providing us with an initial starting point.

We would also like to thank the lab assistants and faculty members of EXTC department, DJ Sanghvi College of Engineering for providing miscellaneous support required.

A special mention of appreciation for our family and friends who provided us moral support needed to sustain the efforts required in the project. Their encouragement helped us in completing the project efficiently.

iii

**ABSTRACT**

An Artificial Neural Network model for air and dielectric suspended circular, semicircular and equilateral triangular Microstrip antenna on thicker air substrate over 800 to 6000 MHz frequency band is proposed. For the substrate thickness varying in the range of 0.04 to 0.1λ0, the resonance frequencies calculated using proposed model agrees closely with the simulated and measured results with error less than 4% over the complete frequency range. This proposed model can be used to design regular shape microstrip patch at any given frequency and on thicker substrate. An efficient configuration of the Neural Network for designing antennas is presented by comparing mean square error values and calculation complexity for different configurations.

iv

**LIST OF TABLES**

Semicircular Air Suspended…………………………………………… A-1

Semicircular Dielectric………………………………………………… A-3

Circular Air Suspended………………………………………………... A-6

Circular Dielectric……………………………………………………… A-9

Equilateral Triangle Air Suspended…………………………………… A-12

Equilateral Triangle Dielectric…………………………………………. A-15

v

**LIST OF FIGURES**

**Figure 1**: Various shapes of patch microstrip antennas…………………………………..4

**Figure 2**: Geometry of circular microstrip antenna……………………………………….5

**Figure 3**: Coaxial Probe feed to patch antenna………………………………………........6 **Figure 4**: ETMSA Geometry……………………………………………………………...7 **Figure 5:** Fringing Field in MSA………………………………………………………….8 **Figure 6:** Multilayer Perceptron…………………………………………………………..16 **Figure 7:** Backpropagation Algorithm……………………………………………………17 **Figure 8:** IE3D simulation………………………………………………………………..19 **Figure 9:** Python Logo……………………………………………………………………21 **Figure 10:** NumPy Logo………………………………………………………………….22 **Figure 11:** Network Input Output………………………………………………………...23 **Figure 12:** Mean square error vs Epoch for various configurations……………………..24 **Figure 13** Flowchart………………………………………………………………………25 **Figure 14:** Testing the neural network for 1000MHz and 27 mm substrate height……...26 **Figure 15**: IE3D simulation for ETMSA……………………………………………….....27 **Figure 16**: IE3D simulation for Circular………………………………………………….27 **Figure 17:** Simulation setup in IE3D……………………………………………………..28 **Figure 18:** Resonance Frequency verification on IE3D………………………………….28

vi

**INTRODUCTION**

Microstrip antennas are being frequently used in wireless applications due to its light weight, low profile, low cost and ease of integration with microwave circuits. Microstrip antennas are used in high-performance applications such as spacecraft, aircraft, missile and satellite, where size, weight, cost, performance, ease of installation and aerodynamic profile are constraints and low profile antennas may be required. To meet these requirements, microstrip antennas offer an optimal solution. Microstrip antennas are also referred to as patch antennas because of the radiating elements (patches) photoetched on the dielectric substrate. This radiating patch may be square, rectangular, circular, elliptical, triangular and any other desired configuration. Neural networks have recently gained attention as a fast and flexible vehicle to EM/microwave modeling, simulations and optimization. ANN can be used efficiently to design of various types microstrip antenna. Although microstrip antennas have proven to be a significant advance in antenna technology, their main drawbacks can be attributed to narrow bandwidth and low gain. The major parts of patch antenna research pro-posed in the literature have basically concentrated on rectangular, triangular and circular microstrip antennas as their regular shapes provide ease of analysis and design. The size of regular shaped patch antennas is relatively big for UHF band applications; hence, their configurations need to be modified at these frequencies. Therefore, size reduction or compactness has been the key design consideration for wire-less communication applications. Hence our proposed project has been designed to predict the effective dimensions for semi-circular microstrip antenna (SCMSA), circular microstrip antenna (CMSA) and equilateral triangle microstrip antenna (ETMSA).

In a comparative study evaluation of different variants of back propagation training algorithm has done for the design of rectangular microstrip antenna. ANN can also be used to calculate different parameters of circular microstrip antenna such as resonant frequency, input impedance etc. Sufficient amount of work indicates how ANN can be used efficiently to design circular and equilateral triangle microstrip antennas. Also ANN can be used to calculate different parameters of circular and equilateral triangle microstrip antenna such as radiation efficiency, resonating frequency, directivity, feed position, resonant frequencies of triangular and rectangular microstrip antennas, resonant resistance calculation of electrically thin and thick rectangular microstrip antennas input impedance of rectangular microstrip

**1**

antennas. However in our proposed project we are only looking for the designing parameters that are the effective dimensions of an antenna which are directly dependent on the radiation field pattern and the fringing field extension.

**1.1 Problem Statement**

As a part of the project we have designed a neural network model implemented in Python 2.7 to estimate the radius of the semi-circular, circular and equilateral triangle microstrip antenna incorporating fringing fields present due to excitation of antenna. Traditional closed loop equations exist for calculating the effective patch radius [1-3]. The formula is dependent on the resonant frequency, effective dielectric of substrate and substrate height. The above formula is valid only for substrate height < 0.04λ.

**1.2 Requirement of ANN**

With increase in substrate height, the above closed form equation is invalid. While methods exist to estimate the effective radius, the calculations involved are complex to perform. We present a neural network model which learns the relation between input theoretical radius obtained by the formula, λ/4 for fundamental mode of operation and the effective patch radius. We also present an analysis to derive the number of hidden layers and the hidden nodes in each layer. The neural network is trained by providing input frequency, effective substrate, theoretical radius and substrate height, with effective radius as its output. We also present the error distribution across frequencies from 800MHz to 6000MHz.

In this project, an artificial neural network (ANN) model for predicting the resonance frequency of modified shape MSA at its fundamental mode is proposed. The ANN model is proposed for varying substrate thickness and over wide frequency range.While calculating the resonance frequency, an extension in patch dimensions due to the fringing fields around patch periphery, is added. The equations for fringing field extension on thinner substrates are reported. However the reported literature does not provide resonance frequency formulation for substrate thickness more than 0.03λ 0 as well as over the wide frequency range. Therefore we propose an ANN model to provide values of side length for substrate thickness greater than 0.03λ 0. The antenna used is a patch Microstrip with air as substrate. We vary the thickness of substrate and patch size to obtain different resonant frequencies.

**2**

**MICROSTRIP ANTENNA**

**2.1 Introduction to Microstrip Antenna**

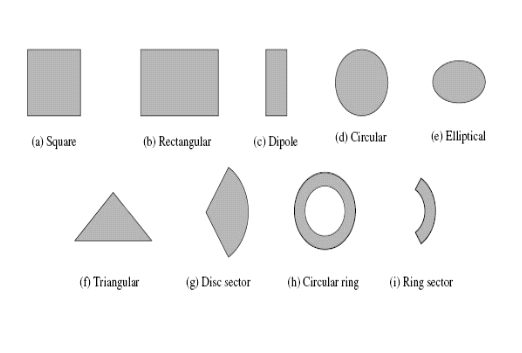
A microstrip patch antenna (also known as a rectangular microstrip antenna) is a type of radio antenna with a low profile, which can be mounted on a flat surface. It consists of a flat rectangular sheet or patch of metal, mounted over a larger sheet of metal called a ground plane. They are the original type of microstrip antennas described by Howell in 1972, with two metal sheets together form a resonant piece of microstrip transmission line with a length of approximately one-half wavelength of the radio waves. The radiation mechanism arises from discontinuities at each truncated edge of the microstrip transmission line. The radiation at the edges causes the antenna to act slightly larger electrically than its physical dimensions, so in order for the antenna to be resonant, a length of microstrip transmission line slightly

shorter than one-half a wavelength at the frequency is used.

Microstrip antennas basically consist of a radiating patch on one side of a dielectric substrate, which has a ground plane on the other side. The patch is generally made of conducting material such as copper and gold. The patch is very thin; λo usually 0.003 λo ≤h ≤ 0.05 λo ) above the ground plane. The microstrip patch is designed so its pattern maximum is normal to the patch (broadside radiator). This is accomplished by properly choosing the mode (field configuration) of excitation beneath the patch. There are numerous substrates that can be used for the design of microstrip patch antennas and their dielectric constants are usually in the range of 2.2 ≤εr≤ 12. Those desirable for antenna performance are thick substrates whose dielectric constant are in the lower end of the range due to better efficiency, larger bandwidth, and loosely bound fields for radiation into space but at the expense of larger element size. Microstrip patch antennas radiate primarily because of the fringing fields between the patch edge and the ground plane. The radiation increases with frequency, thicker substrates, lower permittivity, and originates mostly at discontinuities). Since microstrip antennas are often integrated with other microwave circuitry, a compromise has to be reached between good antenna performance and circuit design. The radiating element and the feed lines are usually photo etched on the dielectric substrate. The radiating patch may be square, rectangle, thin strip (dipole), circular, elliptical, triangle or any other configuration. A microstrip antenna is

**3**

very versatile and made for a wide range of resonant frequencies, polarization patterns and impedances. Due to its operational features viz. low efficiency, low power, high quality factor, poor polarization purity, poor scan performance and very narrow frequency bandwidth, it is suitable for mobile and government security systems where narrow bandwidth are priority. They are also used on laptops, microcomputers, mobile phones etc.



**Figure 1**: Various shapes of patch microstrip antennas

**2.2 Circular Microstrip Antenna**

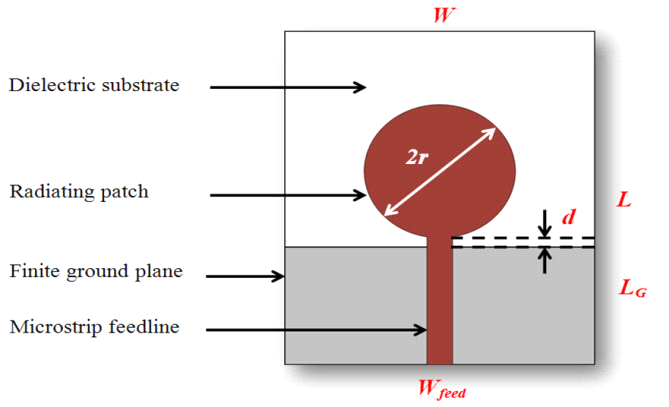
The mode supported by the circular patch antenna can be found by treating the patch, ground plane and the material between the two as a circular cavity. The radius of the patch is the only degree of freedom to control the modes of the antenna (Balanis, 1982). The antenna can be conveniently analysed using the cavity model (Richards, 1988; Gonca, 2005). The cavity is composed of two electric conductors at the top and the bottom to represent the patch and the ground plane and by a cylindrical perfect magnetic conductor around the circular periphery of the cavity. The dielectric material of the substrate is assumed to be truncated beyond the extent of the patch (Richards, 1988). The field configuration within the cavity can be found using the vector potential. The magnetic vector potential Az must satisfy, the homogeneous wave equation (Balanis, 1982) . For a semi-circular microstrip antenna the radiation pattern is a function of Bessel functions as opposed to a rectangular microstrip antenna.

**4**

Electric and Magnetic Fields- To ﬁnd the ﬁelds within the cavity, we use the vector potential approach. For 𝑇𝑀𝑧 we need to ﬁrst ﬁnd the magnetic vector potential 𝐴𝑧, which must satisfy, in cylindrical coordinates, the homogeneous wave equation :-

Δ2Az(ρ,φ,z) + k2Az(ρ,φ,z) = 0

The primed cylindrical coordinates ρ′, φ′, z′are used to represent the ﬁelds within the cavity using Jm(x) which is the Bessel function of the ﬁrst kind of order m.



**Figure 2**: Geometry of circular microstrip antenna

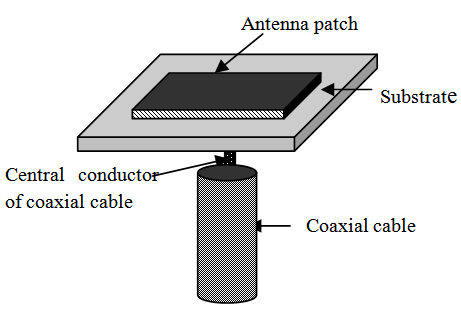
**Feeding Techniques.**

1.Microstrip Line Feed.:- A conducting strip is connected to the edge of the patch. The feed can be etched on the substrate.

2.Capacitive Feeding:- In this type of feeding the feeding is done to small another patch instead of main radiating patch.

3. Coaxial Feeding:- The Coaxial feed or probe feed is a very common technique used for feeding Microstrip patch antennas. The center conductor of the coaxial connecter is soldered to the patch. Advantages are it is easy to fabricate and match as well as low spurious radiation. Disadvantages are narrow bandwidth. We have made use of a probe feed connector to provide feed to the compact microstrip antennas.

**5**



**Figure 3**: Coaxial Probe feed to patch antenna

Advantages :-

1. Low weight and small volume.

2. Low fabrication cost

3. Allows linear and circular polarization.

4. Mechanically robust.

5. Capable of dual and triple frequency operations.

Disadvantages and its remedies:-

1. Low power and low gain can overcome by arrays configuration.

2. Surface wave associated limitations such as poor efficiency, increased mutual coupling, reduced gain and radiation pattern can be overcome.

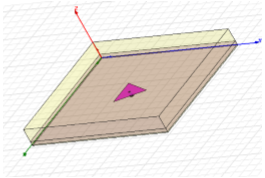
3. The band width can increase up to 60% by using Bandwidth Enhancement Techniques.

**2.3 Equilateral Microstrip Antenna**

The triangular geometry of microstrip antenna is one of the most common shapes having a wide range of wireless application ranging from circuit element to wireless antennas. Compact microstrip antennas have recently received much attention due to the increasing demand of small antennas for personal as well as commercial communication equipment. It has been demonstrated that equilateral triangular microstrip patch can effectively reduce the required patch size for a given operating frequency. In mobile communication system such as satellite, RADAR, Global Position System (GPS) often require extremely small size, light

**6**

weight. The ‘C’ band of frequency are used for the satellite communication and terrestrial application. Single band & Dual band frequency operation of triangular microstrip antennas have been studied by many researchers using coaxial probe feed. This project reports the simulation result using equilateral triangular patch antenna with co-axial feed. The geometry of the proposed triangular antenna using a co-axial probe feed is shown. The proposed antenna is constructed on a dielectric substrate on different thickness such as:-



**Figure 4**: ETMSA Geometry

Some of the principal advantages of microstrip equilateral triangle patch antenna discussed are: Lightweight and low volume, low profile planar configuration which can be easily made conformal to host surface. Low fabrication cost, hence can be manufactured in large quantities, supports both, linear as well as circular polarization, capable of dual and triple frequency operations, mechanically robust when mounted on rigid surfaces. In spite of the many advantages, these antennas also suffer from a number of disadvantages. Some of them are : Narrow bandwidth and Low efficiency, Low gain, Extraneous radiation from feeds and junctions. Poor end fire radiator except tapered slot antennas, Low power handling capacity and Surface wave excitation.

Microstrip patch antennas have a very high antenna quality factor (Q). Q represents the losses associated with the antenna. Typically there are radiations, conduction (ohmic), dielectric and surface wave losses. For very thin substrates, the losses due to surface waves are very small and can be neglected. However, as the thickness increases, an increasing fraction of the total power delivered by the source goes into a surface wave. This surface wave contribution is

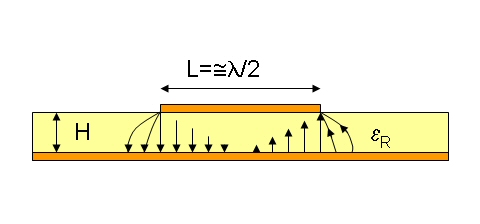
**7**

considered as an unwanted power loss since it is ultimately scattered at the dielectric bends and causes degradation of the antenna characteristics. The surface waves can be minimized by use of photonic bandgap structures as discussed. Other problems such as lower gain and lower power handling capacity can be overcome by using an array configuration for the elements.

**2.4 Fringing Fields in ANN**

The fringing fields around the antenna can help explain why the microstrip antenna radiates. Consider the side view of a patch antenna, shown in Figure 5. Note that since the current at the end of the patch is zero (open circuit end), the current is maximum at the center of the half-wave patch and (theoretically) zero at the beginning of the patch. This low current value at the feed explains in part why the impedance is high when fed at the end (we'll address this again later).

Since the patch antenna can be viewed as an open circuited transmission line, the voltage reflection coefficient will be 1 (see the transmission line tutorial for more information). When this occurs, the voltage and current are out of phase. Hence, at the end of the patch the voltage is at a maximum (say +V volts). At the start of the patch antenna (a half-wavelength away), the voltage must be at minimum (-V Volts). Hence, the fields underneath the patch will resemble that of Figure 5, which roughly displays the fringing of the fields around the edges.



**Figure 5:** Fringing Field in MSA

**8**

It is the fringing fields that are responsible for the radiation. Note that the fringing fields near the surface of the patch antenna are both in the +y direction. Hence, the fringing E-fields on the edge of the microstrip antenna add up in phase and produce the radiation of the microstrip antenna. This paragraph is critical to understanding the patch antenna. The current adds up in phase on the patch antenna as well; however, an equal current but with opposite direction is on the ground plane, which cancels the radiation. This also explains why the microstrip antenna radiates but the microstrip transmission line does not. The microstrip antenna's radiation arises from the fringing fields, which are due to the advantageous voltage distribution; hence the radiation arises due to the voltage and not the current. The patch antenna is therefore a "voltage radiator", as opposed to the wire antennas, which radiate because the currents add up in phase and are therefore "current radiators".

As a side note, the smaller epsilon is, the more "bowed" the fringing fields become; they extend farther away from the patch. Therefore, using a smaller permittivity for the substrate yields better radiation. In contrast, when making a microstrip transmission line (where no power is to be radiated), a high value of epsilon is desired, so that the fields are more tightly contained (less fringing), resulting in less radiation. This is one of the trade-offs in patch antenna design. There have been research papers written were distinct dielectrics (different permittivities) are used under the patch antenna and transmission line sections, to circumvent this issue.

**9**

**ARTIFICIAL NEURAL NETWORKS**

**3.1 Introduction to ANN**

Artificial neural networks (ANNs) or connectionist systems are a computational model used in machine learning, computer science and other research disciplines, which is based on a large collection of connected simple units called artificial neurons, loosely analogous to axons in a biological brain. Connections between neurons carry an activation signal of varying strength.[further explanation needed] If the combined incoming signals are strong enough, the neuron becomes activated and the signal travels to other neurons connected to it. Such systems can be trained from examples, rather than explicitly programmed, and excel in areas where the solution or feature detection is difficult to express in a traditional computer program. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks, like computer vision and speech recognition, that are difficult to solve using ordinary rule-based programming.

Typically, neurons are connected in layers, and signals travel from the first (input), to the last (output) layer. Modern neural network projects typically have a few thousand to a few million neural units and millions of connections; their computing power is similar to a worm brain, several orders of magnitude simpler than a human brain. The signals and state of artificial neurons are real numbers, typically between 0 and 1. There may be a threshold function or limiting function on each connection and on the unit itself, such that the signal must surpass the limit before propagating. Back propagation is the use of forward stimulation to modify connection weights, and is sometimes done to train the network using known correct outputs. However, the success is unpredictable: after training, some systems are good at solving problems while others are not. Training typically requires several thousand cycles of interaction.

The goal of the neural network is to solve problems in the same way that a human would, although several neural network categories are more abstract. New brain research often stimulates new patterns in neural networks. One new approach is use of connections which

**10**

span further to connect processing layers rather than adjacent neurons. Other research being explored with the different types of signal over time that axons propagate, such as deep learning, interpolates greater complexity than a set of boolean variables being simply on or off. Newer types of network are more free flowing in terms of stimulation and inhibition, with connections interacting in more chaotic and complex ways. Dynamic neural networks are the most advanced, in that they dynamically can, based on rules, form new connections and even new neural units while disabling others.

Historically, the use of neural network models marked a directional shift in the late 1980s from high-level (symbolic) artificial intelligence, characterized by expert systems with knowledge embodied in if-then rules, to low-level (sub-symbolic) machine learning, characterized by knowledge embodied in the parameters of a cognitive model with some dynamical system.

**3.2 Types of ANN**

The various types of neural network are :

1) Feedforward Neural Network

The feedforward neural network was the first and arguably most simple type of artificial neural network devised. In this network the information moves in only one direction—forward: From the input nodes data goes through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. Feedforward networks can be constructed from different types of units, e.g. binary McCulloch-Pitts neurons, the simplest example being the perceptron. Continuous neurons, frequently with sigmoidal activation, are used in the context of backpropagation of error.

2) Radial Basis Function

Radial basis functions are powerful techniques for interpolation in multidimensional space. A RBF is a function which has built into it a distance criterion with respect to a center. Radial basis functions have been applied in the area of neural networks where they may be used as a replacement for the sigmoidal hidden layer transfer characteristic in multi-layer perceptrons. RBF networks have two layers of processing: In the first, input is mapped onto each RBF in

**11**

the 'hidden' layer. The RBF chosen is usually a Gaussian. In regression problems the output layer is then a linear combination of hidden layer values representing mean predicted output. The interpretation of this output layer value is the same as a regression model in statistics. In classification problems the output layer is typically a sigmoid function of a linear combination of hidden layer values, representing a posterior probability. Performance in both cases is often improved by shrinkage techniques, known as ridge regression in classical statistics and known to correspond to a prior belief in small parameter values (and therefore smooth output functions) in a Bayesian framework.

3) Associative Neural Network

The ASNN is an extension of the committee of machines that goes beyond a simple/weighted average of different models. ASNN represents a combination of an ensemble of feedforward neural networks and the k-nearest neighbor technique (kNN). It uses the correlation between ensemble responses as a measure of distance amid the analyzed cases for the kNN. This corrects the bias of the neural network ensemble. An associative neural network has a memory that can coincide with the training set. If new data become available, the network instantly improves its predictive ability and provides data approximation (self-learn the data) without a need to retrain the ensemble. Another important feature of ASNN is the possibility to interpret neural network results by analysis of correlations between data cases in the space of models

4) Recurrent Neural Network (Hopfield/SOM)

Contrary to feedforward networks, recurrent neural networks (RNNs) are models with bi-directional data flow. While a feedforward network propagates data linearly from input to output, RNNs also propagate data from later processing stages to earlier stages. RNNs can be used as general sequence processors. The Hopfield network (like similar attractor-based networks) is of historic interest although it is not a general RNN, as it is not designed to process sequences of patterns. Instead it requires stationary inputs. It is an RNN in which all connections are symmetric. Invented by John Hopfield in 1982 it guarantees that its dynamics will converge. If the connections are trained using Hebbian learning then the Hopfield

**12**

network can perform as robust content-addressable memory, resistant to connection alteration. The self-organizing map (SOM) invented by Teuvo Kohonen performs a form of unsupervised learning. A set of artificial neurons learn to map points in an input space to coordinates in an output space. The input space can have different dimensions and topology from the output space, and the SOM will attempt to preserve these.

* 1. **Learning Mechanism**

There are three major learning paradigms, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning.

1) Supervised Learning

In [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), we are given a set of example pairs ( x , y ) , x ∈ X , y ∈ Y {\displaystyle (x,y),x\in X,y\in Y} and the aim is to find a function f : X → Y {\displaystyle f:X\rightarrow Y} in the allowed class of functions that matches the examples. In other words, we wish to *infer* the mapping implied by the data; the cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain.

A commonly used cost is the [mean-squared error](https://en.wikipedia.org/wiki/Mean-squared_error), which tries to minimize the average squared error between the network's output, f ( x ) {\displaystyle f(x)} , and the target value y {\displaystyle y} over all the example pairs. When one tries to minimize this cost using [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent) for the class of neural networks called [multilayer perceptrons](https://en.wikipedia.org/wiki/Multilayer_perceptron) (MLP), one obtains the common and well-known [backpropagation algorithm](https://en.wikipedia.org/wiki/Backpropagation) for training neural networks.

Tasks that fall within the paradigm of supervised learning are [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition) (also known as classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis) (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher", in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

2) Unsupervised learning

In unsupervised learning, some data (x {\displaystyle \textstyle x} x) is given and the cost function to be minimized, that can be any function of the data (x {\displaystyle \textstyle x} x) and the network's output(f {\displaystyle \textstyle f} f).The cost function is dependent

**13**

on the task (what we are trying to model) and our *a priori* assumptions (the implicit properties of our model, its parameters and the observed variables).

As a trivial example, consider the model f ( x ) = a {\displaystyle \textstyle f(x)=a} f(x)=m where a {\displaystyle \textstyle a} m is a constant and the cost C = E [ ( x − f ( x ) ) 2 ] {\displaystyle \textstyle C=E[(x-f(x))^{2}]} C=E[(x-f(x))2]. Minimizing this cost will give us a value of a {\displaystyle \textstyle a} m that is equal to the mean of the data. The cost function can be much more complicated. Its form depends on the application: for example, in compression it could be related to the mutual information between x x {\displaystyle \textstyle x} and f(x) f ( x ) {\displaystyle \textstyle f(x)} , whereas in statistical modeling, it could be related to the posterior probability of the model given the data (note that in both of those examples those quantities would be maximized rather than minimized).Tasks that fall within the paradigm of unsupervised learning are in general estimation problems; the applications include clustering, the estimation of statistical distributions, compression and filtering.

3) Reinforcement Learning

In reinforcement learning, data x {\displaystyle \textstyle x} x are usually not given, but generated by an agent's interactions with the environment. At each point in time t {\displaystyle \textstyle t} t, the agent performs an action y t {\displaystyle \textstyle y\_{t}} ytand the environment generates an observation xt x t {\displaystyle \textstyle x\_{t}} and an instantaneous cost c t {\displaystyle \textstyle c\_{t}} ct, according to some (usually unknown) dynamics. The aim is to discover a *policy* for selecting actions that minimizes some measure of a long-term cost, e.g., the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated. ANNs are frequently used in reinforcement learning as part of the overall algorithm. Dynamic programming has been coupled with ANNs (giving neurodynamic programming) by Bertsekas and Tsitsiklis and applied to multi-dimensional nonlinear problems such as those involved in vehicle routing, natural resources management or medicine because of the ability of ANNs to mitigate losses of accuracy even when reducing the discretization grid density for numerically approximating the solution of the original control problems.Tasks that fall within the paradigm of reinforcement learning are control problems, games and other sequential decision making tasks.

**14**

* 1. **Multi-Layer Perceptron**

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network. MLP is a modification of the standard linear perceptron and can distinguish data that is not linearly separable.

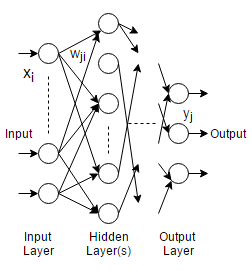
Activation function:If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then it is easily proven with linear algebra that any number of layers can be reduced to the standard two-layer input-output model (see perceptron). What makes a multilayer perceptron different is that some neurons use a nonlinear activation function which was developed to model the frequency of action potentials, or firing, of biological neurons in the brain. This function is modeled in several ways.

The two main activation functions used in current applications are both sigmoids, and are described by



in which the former function is a hyperbolic tangent which ranges from -1 to 1, and the latter, the logistic function, is similar in shape but ranges from 0 to 1. Here y is the output of the node (neuron) and v is the weighted sum of the input synapses. Alternative activation functions have been proposed, including the rectifier and softplus functions. More specialized activation functions include radial basis functions which are used in another class of supervised neural network models.

Layers: The multilayer perceptron consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes and is thus considered a deep neural network. Since an MLP is a Fully Connected Network, each node in one layer connects with a certain weight to every node in the following layer



**Figure 6:** Multilayer Perceptron

* 1. **Backpropagation Algorithm**

The Backpropagation algorithm is a Supervised Learning algorithm which reduces error by changing the Weights along direction of steepest gradient. The weights closest to the output layer are altered followed by the weights closer to the Input layer. To calculate weight change, the algorithm uses Mean Square Error between the Target output and the network output which is given by

The change in weight is a function of the Learning Rate η which determines the rate of convergence and is chosen between 0.5 and 1.0. Another term which determines the contribution of previous weights is momentum factor α which increases the rate of convergence and its value is chosen as 0.9. The updation of weights is given by:

The error correction term δ depends on the derivative of the activation function chosen and is expressed by:

Training of the network continues till either the epoch limit is reached or the error over all iterations in an epoch is smaller than the pre-decided threshold error value.

**16**

initialize network weights (often small random values)

**do**

**forEach** training example named ex

prediction = neural-net-output(network, ex) actual = teacher-output(ex)

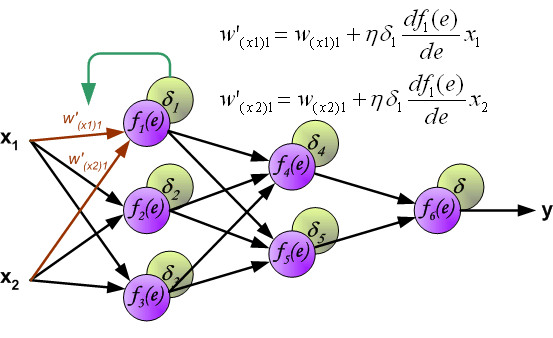
compute error (prediction - actual) compute Δ w h {\displaystyle \Delta w\_{h}} for all weights from hidden layer to output layer *// backward pass*

compute Δ w i {\displaystyle \Delta w\_{i}} for all weights from input layer to hidden *// backward pass continued*

update network weights *// input layer not modified by error estimate*

**until** all examples classified correctly or another stopping criterion satisfied

**return** the network



**Figure 7:** Backpropagation Algorithm

**17**

**IE3D**

**4.1 Introduction**

IE3D from Mentor Graphics, formerly Zeland Software is the first scalable EM design and verification platform that delivers the modeling accuracy for the combined needs of high-frequency circuit design and signal integrity engineers across multiple design domains.

For many companies, there is no longer just one EM problem at hand, but several different ones each presenting a unique bottleneck and delaying overall design closure. IE3D's multi-threaded and distributed simulation architecture and high-design capacity is the most cost-effective EM simulation and modeling solution for component-level and circuit-level applications. IE3D offers the highest simulation capacities and fastest turnaround times for the broadest number of applications making it the best choice for improving your design team productivity and meeting design schedules on time.

**IE3D for Antenna Design**

Today's high performance antenna array design requires both, large capacity EM simulation and unit array cell EM design and optimization capabilities. Getting the array unit cell right from the start is essential before replicating into a larger antenna array structure. And if accuracy is sought, then the designer needs to be extremely careful with approaches that formulate estimated boundary conditions for unit cells used within larger arrays. These approaches typically suffer from poor capacity limits and do not accurately model the EM behavior between unit cells, especially for cells on the antenna array periphery. IE3D-SSD offers FASTEM to thoroughly explore the relevant design space and optimize the geometry for each unique unit cell. In addition, IE3D-SSD's superior capacity and run-time enable even the largest antenna arrays to be solved in least amount of time. IE3D-SSD is the best solution for your antenna design.

**18**

**4.2 Simulation Using IE3D**

In order to train the neural network over a range of frequencies we have first gathered the dataset which can be used for training our neural network model. To generate the target output values of practical radius values we have simulated the semi-circular microstrip antenna in real conditions using IE3D software.

Collection of dataset:-

For a particular frequency we :

1)Vary the substrate thickness(h) over a range of 0.04\*λ <h<0.1\*λ

2)Adjust radius to get desired resonance frequency curve

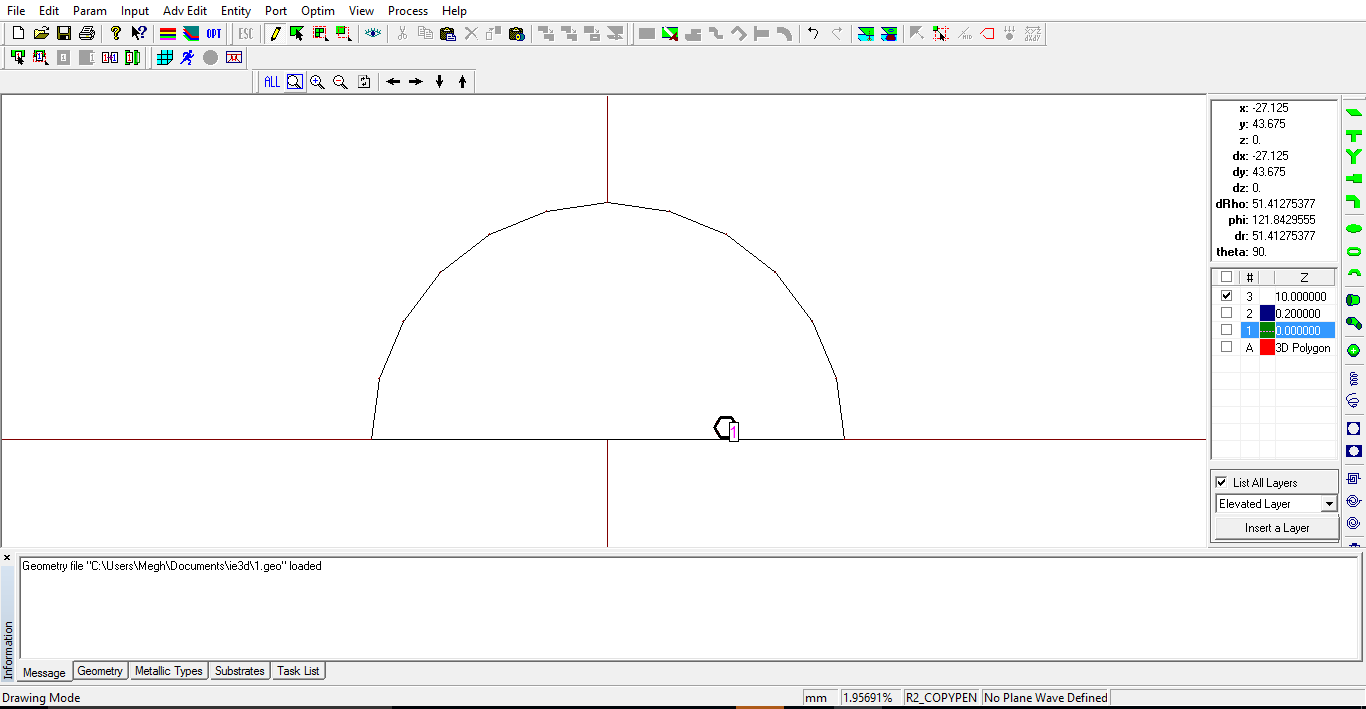
This radius (a) takes into account the fringing field effects of compact MSA. The difference between theoretical effective radius (ae) and practical patch radius (a) is used to obtain the amount of fringing (Δa).The theoretical radius can be calculated using formula:-

ae = 1.8411\*30 ; for fundamental mode of operation.

2Π \* fr \*√(ɛr)

also, ae=a(prac) + Δa

Thus the practical radius should always be lesser than the actual theoretical radius as it accounts for the fringing field extension.Data has been collected from 800 Mhz upto 6000 Mhz and this data has been fed to the neural network for training.



**Figure 8:** IE3D simulation

**19**

**NEURAL NETWORK PROGRAMMING**

**5.1 Python**

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale.

Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Python is a multi-paradigm programming language: object-oriented programming and structured programming are fully supported, and many language features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a mix of reference counting and a cycle-detecting garbage collector for memory management. An important feature of Python is dynamic name resolution (late binding), which binds method and variable names during program execution.

The design of Python offers some support for functional programming in the Lisp tradition. The language has map(), reduce() and filter() functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

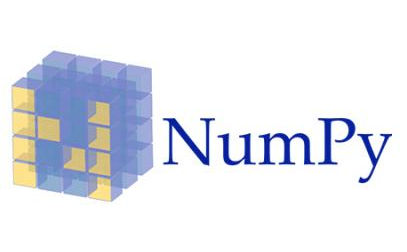
**20**



**Figure 9:** Python Logo

**5.2 NumPy**

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.



**Figure 10:** NumPy Logo

NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy address the slowness problem partly by

**21**

providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring (re)writing some code, mostly inner loops using NumPy.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted,[3] and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

**22**

**NEURAL NETWORK ARCHITECTURE**

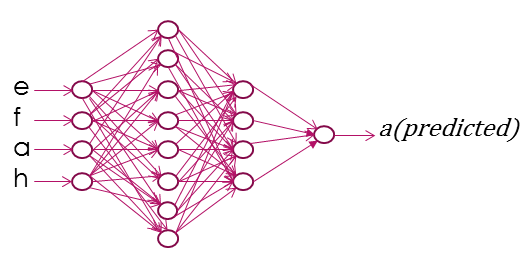
**6.1 Network Parameters**

The architecture of the neural network proposed is the Multilayer Perceptron, which is a feedforward network consisting of Input layer, Hidden Layers and Output layers. The Learning Algorithm employed is the Backpropagation algorithm, which is a variation of the Steepest Gradient method of calculating the Weights of the network.

Initially, Weights of the network are initialized randomly between -1 and 1. For better prediction, the inputs and outputs are normalized between 0 and 1. The input of the multilayer perceptron acts as a buffer and has linear activation function. The outputs of the input neurons undergo weighted distribution to the inputs of the hidden layers. This is followed by summation of inputs and application of an activation function which is given by:

**The values of sigmoidal gain, momentum rate and threshold are chosen as 0.6, 0.9 and 0 respectively**. The input training data set is generated by considering frequencies from 800 MHz to 6000 MHz at intervals of 500 MHz. For each frequency, the height of the dielectric is varied from 0.04λ to 0.1λ in steps of 0.01λ.

**6.2 Network I/O**

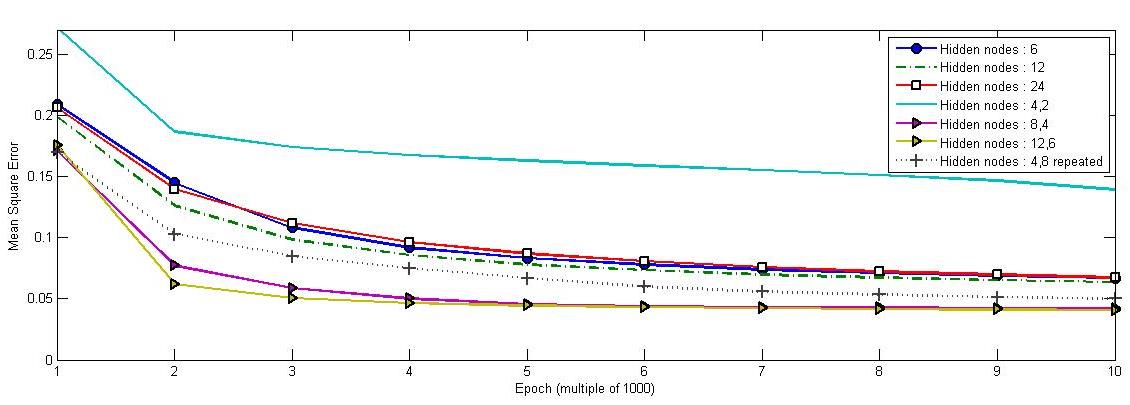


**Figure 11:** Network Input Output

The proposed neural network model is implemented to estimate the patch radius for the air suspended and dielectric Circular and Semi-Circular Microstrip Antenna. Predicted patch radius includes the effect of fringing field present. Inputs to the network are the Resonant Frequency, Dielectric constant (), Quarter Wavelength () and Height of Substrate The network has 2 hidden layers followed by a one neuron output layer. The output of the network is the predicted radius. The input layer of the ANN has linear activation function while the hidden and output layers have a sigmoidal activation function given by:

**6.3 Configuration of Hidden Layer(s)**

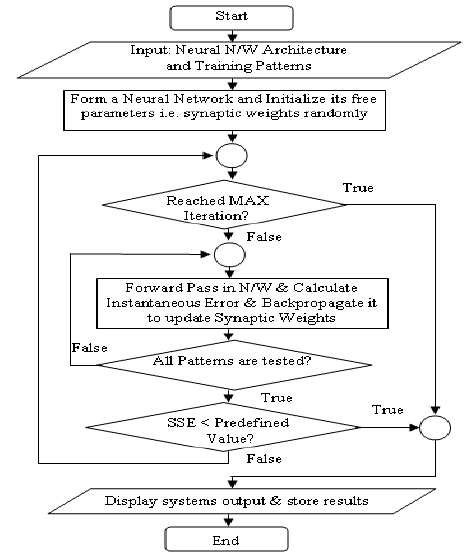
To evaluate the optimum number of hidden layers and the hidden neurons in each layer, the network was simulated for various configurations. Mean Square Errors for each configuration were calculated at different epoch values. A comparison of these errors is shown in Fig 3. It is observed that the error for a single layer hidden neuron is greater than the error generated by a network consisting of two hidden layers. It is also observed that the performance of the network converges to a finite non-zero value as the number of epoch increases. Inferring from the plot, the appropriate configuration is a network consisting of 8,4 neurons in each hidden layer. Other configurations such as 12,6 were rejected as they increased network complexity while yielding no appreciable performance improvement.



**Figure 12:** Mean square error vs Epoch for various configurations

**24**

**6.4 Training**

**Figure 13:** Flowchart

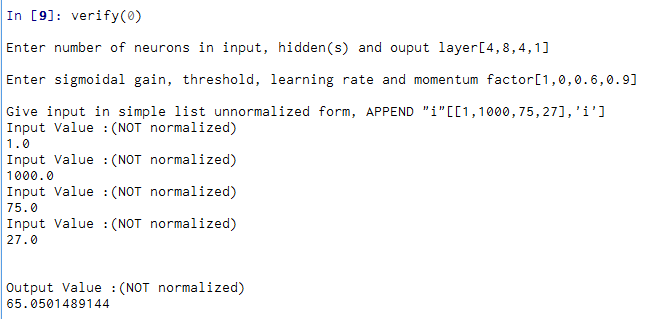
The network is trained for 10,000 epochs with error margins obtained for output is less than 4% (barring few exceptions). The network trains itself with the data set generated using the IE3D simulation software. The network is trained over a particular frequency and height range and the output radius can be predicted for a much wider range .

In addition, the use of neural network in designing of MSA reduces the need for complex computations and calculation. The ability of the neural network to learn the trends is used to reduce the steps required and thus speed up the entire designing process with additional

**25**

accuracy. It can be observed that the designing process is faster using neural networks compared to the conventional methods.

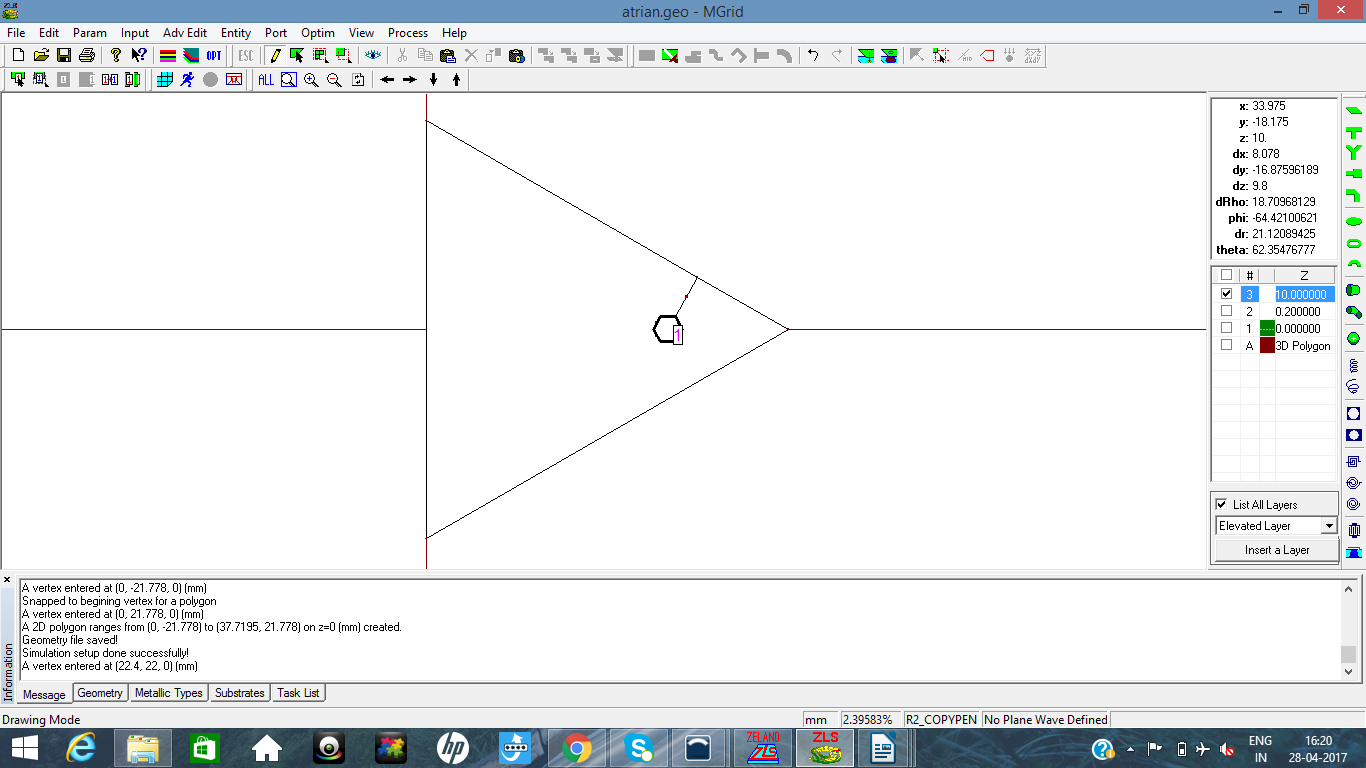
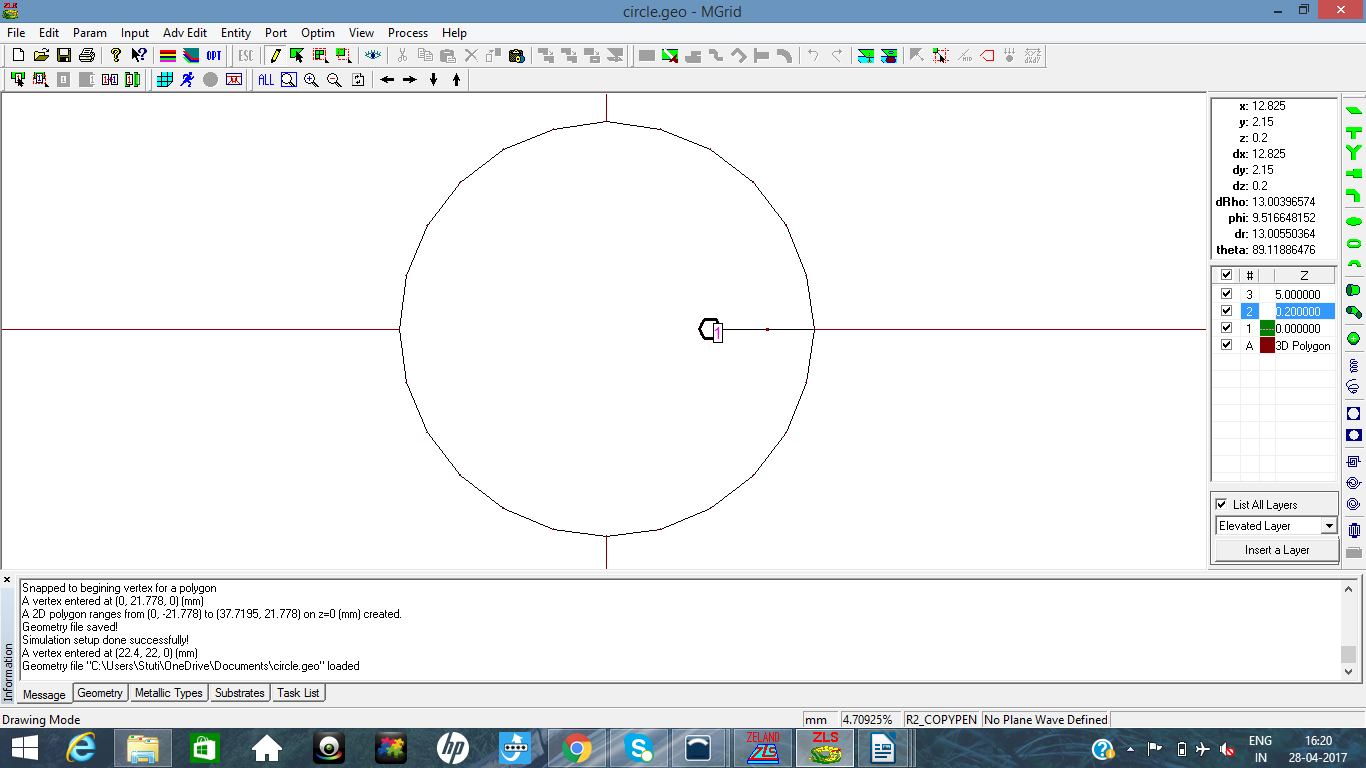
**6.5 Testing**



**Figure 14:** Testing the neural network for 1000MHz and 27 mm substrate height

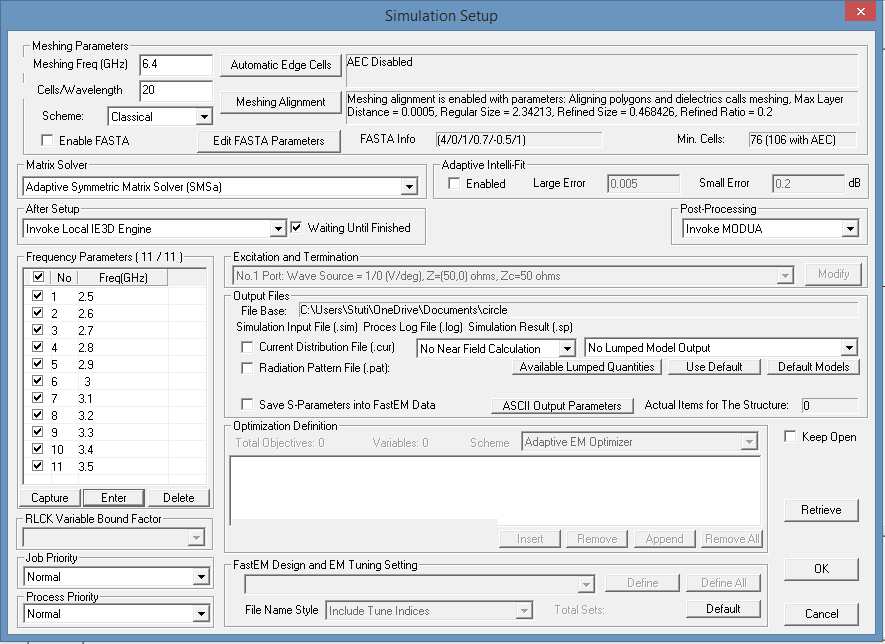
The testing data set and corresponding radius and error values are reported in Table (x-y). Error for Air suspended MSAs were found to be consistently lower than those obtained for Dielectric suspended MSAs. Higher errors were reported with increase in frequencies. The errors at extreme values of dielectric thickness ( and 0.1λ) are relatively greater than errors at mid values (0.06λ to 0.09λ) due to reduced training data at the extreme values. The network programming is done in Python 2.7 with the code supporting adjustable values of network parameters and variable number of layers. (The first author can be contacted via electronic mail to obtain the source code.) . For the implemented neural network model we have trained the model for 10,000 epochs. Testing of the neural network is performed over a frequency range of 800 Mhz to 6000Mhz. Testing is carried out in IE3D

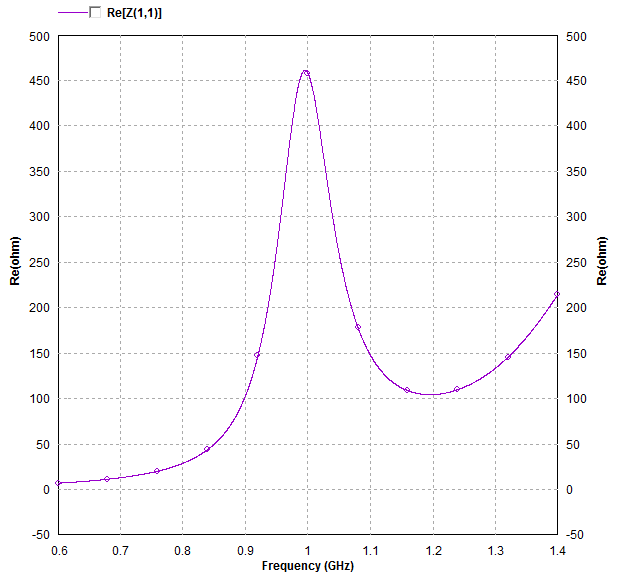
**26**

**Figure 15**: IE3D simulation for ETMSA

**Figure 16**: IE3D simulation for Circular

**27**

**Figure 17:** Simulation setup in IE3D



**Figure 18:** Resonance Frequency verification on IE3D

**CONCLUSION**

The ANN model for ETMSA, CMSA and SCMSA over a wide frequency range and substrate thickness increasing from 0.04 to 0.1λ0, is proposed. The neural network model is developed using antenna parameters like substrate thickness, dielectric constant (air), resonance frequency and an edge extension length in terms of substrate thickness. The training data sets spaced at every 400 MHz frequency intervals were used to develop ANN model. The predicted shorted patch radius as obtained from the neural network model which when simulated using IE3D software gives closer match with the desired patch resonance frequency, over increasing substrate thickness. To validate the simulated and predicted frequency results, measurements were carried out. The measured results show closer agreement with predicted and simulated frequencies. As in the world of miniaturization, MSAs are frequently needed in various applications. In many applications variations of CMSA, i.e. ETMSA, CMSA and SCMSA finds the application. As the close form expressions to calculate shorted patch radius is not available, the proposed ANN model can be used to design Equilateral Triangle, Circular and Semi Circular MSA and on thicker substrate and at any given frequency. In the future work, similar neural network model will be developed to predict the length of shorted MSAs on suspended dielectric substrates. The ANN outputs are in close agreement with the practical values.

**29**

**APPENDIX A (Result Tables)**

**A-1 SCMSA**

**A-1.1 Air**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 12 | 0.04 | 1012 | 1000 | 74.208237 | 1.2 |
| 2 | 15 | 0.05 | 1006.4 | 1000 | 72.393397 | 0.64 |
| 3 | 18 | 0.06 | 1004 | 1000 | 70.548362 | 0.4 |
| 4 | 21 | 0.07 | 1004 | 1000 | 68.695078 | 0.4 |
| 5 | 24 | 0.08 | 998.4 | 1000 | 66.855396 | 0.16 |
| 6 | 27 | 0.09 | 1002.4 | 1000 | 65.050148 | 0.24 |
| 7 | 30 | 0.1 | 1002.4 | 1000 | 63.298240 | 0.24 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 6.00 | 0.04 | 1988 | 2000 | 37.619422 | 0.6 |
| 2 | 7.5 | 0.05 | 1980.0 | 2000 | 36.306375 | 0.96 |
| 3 | 9.00 | 0.06 | 1989.2 | 2000 | 35.014647 | 0.54 |
| 4 | 10.50 | 0.07 | 1997.6 | 2000 | 33.749518 | 0.12 |
| 5 | 12.00 | 0.08 | 2009.6 | 2000 | 32.515841 | 0.48 |
| 6 | 13.50 | 0.09 | 2008.4 | 2000 | 31.317962 | 0.42 |
| 7 | 15 | 0.1 | 2015.6 | 2000 | 30.159662 | 0.78 |

**A-1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 4.00 | 0.04 | 3010 | 3000 | 24.634012 | 0.33 |
| 2 | 5.00 | 0.05 | 2987.2 | 3000 | 23.741410 | 0.42 |
| 3 | 6.00 | 0.06 | 2998.4 | 3000 | 22.871310 | 0.05 |
| 4 | 7.00 | 0.07 | 3006.4 | 3000 | 22.025082 | 0.21 |
| 5 | 8.00 | 0.08 | 3016 | 3000 | 21.203905 | 0.53 |
| 6 | 9.00 | 0.09 | 2998.4 | 3000 | 20.408763 | 0.05 |
| 7 | 10.00 | 0.1 | 3006.4 | 3000 | 19.640443 | 0.21 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 3.00 | 0.04 | 4068.8 | 4000 | 18.095068 | 1.72 |
| 2 | 3.75 | 0.05 | 4025.6 | 4000 | 17.467054 | 0.64 |
| 3 | 4.50 | 0.06 | 4016 | 4000 | 16.856045 | 0.40 |
| 4 | 5.25 | 0.07 | 4017.6 | 4000 | 16.262405 | 0.44 |
| 5 | 6.00 | 0.08 | 4017.6 | 4000 | 15.686418 | 0.44 |
| 6 | 6.75 | 0.09 | 4011.2 | 4000 | 15.128293 | 0.28 |
| 7 | 7.50 | 0.1 | 3969.6 | 4000 | 14.588160 | 0.76 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 2.40 | 0.04 | 5134 | 5000 | 14.284612 | 2.68 |
| 2 | 3.00 | 0.05 | 5064 | 5000 | 13.820683 | 1.28 |
| 3 | 3.60 | 0.06 | 5028 | 5000 | 13.369151 | 0.56 |
| 4 | 4.20 | 0.07 | 5010 | 5000 | 12.930092 | 0.20 |
| 5 | 4.80 | 0.08 | 4994 | 5000 | 12.503551 | 0.12 |
| 6 | 5.62 | 0.09 | 4940 | 5000 | 11.940864 | 1.20 |
| 7 | 6.25 | 0.1 | 4912 | 5000 | 11.524401 | 1.76 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(MHz) | fANN(MHz) | a(mm) Practical | Error (%) |
| 1 | 2.00 | 0.04 | 6192 | 6000 | 11.866323 | 3.2 |
| 2 | 2.5 | 0.05 | 6070 | 6000 | 11.508569 | 1.16 |
| 3 | 3.00 | 0.06 | 6020 | 6000 | 11.159958 | 0.33 |
| 4 | 3.75 | 0.07 | 5980 | 6000 | 10.654164 | 0.33 |
| 5 | 4.00 | 0.08 | 5940 | 6000 | 10.490120 | 1.00 |
| 6 | 4.50 | 0.09 | 5902 | 6000 | 10.168831 | 1.63 |
| 7 | 5.00 | 0.1 | 5824 | 6000 | 9.8565586 | 2.93 |

**A-1.2 Dielectric**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 12 | 0.04 | 1000 | 1000 | 1.1139896 | 66.624292 | 0 |
| 2 | 15 | 0.05 | 1000 | 1000 | 1.089159 | 64.740490 | 0 |
| 3 | 18 | 0.06 | 1001.6 | 1000 | 1.073211 | 62.863304 | 0.16 |
| 4 | 21 | 0.07 | 1001.6 | 1000 | 1.062103 | 60.987453 | 0.16 |
| 5 | 24 | 0.08 | 1003.2 | 1000 | 1.053922 | 59.110960 | 0.32 |
| 6 | 27 | 0.09 | 1006.4 | 1000 | 1.047645 | 57.233357 | 0.64 |
| 7 | 30 | 0.1 | 1011.2 | 1000 | 1.042677 | 55.355011 | 1.12 |

**A-3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 6.0 | 0.04 | 1988.8 | 2000 | 1.25731 | 30.484676 | 0.56 |
| 2 | 7.5 | 0.05 | 2001.6 | 2000 | 1.195773 | 29.471207 | 0.08 |
| 3 | 9.0 | 0.06 | 2001.6 | 2000 | 1.157989 | 28.560792 | 0.08 |
| 4 | 10.5 | 0.07 | 1996.8 | 2000 | 1.13243 | 27.708607 | 0.16 |
| 5 | 12.0 | 0.08 | 1992 | 2000 | 1.113990 | 26.894947 | 0.4 |
| 6 | 13.5 | 0.09 | 1980.8 | 2000 | 1.100057 | 26.109901 | 0.96 |
| 7 | 15.0 | 0.1 | 1966.4 | 2000 | 1.089159 | 25.348035 | 1.68 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 4.00 | 0.04 | 3003.2 | 3000 | 1.44295 | 18.630767 | 0.10667 |
| 2 | 5.00 | 0.05 | 3036.8 | 3000 | 1.32552 | 17.898114 | 1.22667 |
| 3 | 6.00 | 0.06 | 3046.4 | 3000 | 1.25731 | 17.329161 | 1.54667 |
| 4 | 7.00 | 0.07 | 3020.8 | 3000 | 1.21273 | 16.841479 | 0.69333 |
| 5 | 8.00 | 0.08 | 2988.8 | 3000 | 1.18132 | 16.401611 | 0.37333 |
| 6 | 9.00 | 0.09 | 2960 | 3000 | 1.157989 | 15.993355 | 1.3333 |
| 7 | 10.00 | 0.1 | 2926.4 | 3000 | 1.139979 | 15.607954 | 2.45333 |

**A-4**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 3.00 | 0.04 | 3976 | 4000 | 1.25731 | 30.484676 | 0.56 |
| 2 | 3.75 | 0.05 | 4102.4 | 4000 | 1.195773 | 29.471207 | 0.08 |
| 3 | 4.50 | 0.06 | 4097.6 | 4000 | 1.157989 | 28.560792 | 0.08 |
| 4 | 5.25 | 0.07 | 4064 | 4000 | 1.13243 | 27.708607 | 0.16 |
| 5 | 6.00 | 0.08 | 4017.6 | 4000 | 1.113990 | 26.894947 | 0.4 |
| 6 | 6.75 | 0.09 | 3953.6 | 4000 | 1.100057 | 26.109901 | 0.96 |
| 7 | 7.50 | 0.1 | 3889.6 | 4000 | 1.089159 | 25.348035 | 1.68 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 2.40 | 0.04 | 4830 | 5000 | 2.04762 | 10.217801 | 3.4 |
| 2 | 3.00 | 0.05 | 5096 | 5000 | 1.69291 | 9.5251235 | 1.92 |
| 3 | 3.60 | 0.06 | 5128 | 5000 | 1.51765 | 9.1425331 | 2.56 |
| 4 | 4.20 | 0.07 | 5104 | 5000 | 1.41315 | 8.8782284 | 2.08 |
| 5 | 4.80 | 0.08 | 5027.2 | 5000 | 1.34375 | 8.6719310 | 0.544 |
| 6 | 5.62 | 0.09 | 4905.6 | 5000 | 1.27957 | 8.4408588 | 1.888 |
| 7 | 6.25 | 0.1 | 4808 | 5000 | 1.2445 | 8.2873680 | 3.84 |

**A-5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | Error (%) |
| 1 | 2.5 | 0.05 | 5984 | 6000 | 1.96527 | 7.8267804 | 0.26667 |
| 2 | 3.00 | 0.06 | 6102 | 6000 | 1.69291 | 7.4424872 | 1.7 |
| 3 | 3.75 | 0.07 | 6032 | 6000 | 1.48686 | 7.1155062 | 0.53333 |
| 4 | 4.00 | 0.08 | 5974 | 6000 | 1.44295 | 7.0361572 | 0.43333 |
| 5 | 4.50 | 0.09 | 5876 | 6000 | 1.3753 | 6.9012927 | 2.06667 |
| 6 | 5.00 | 0.1 | 5758 | 6000 | 1.32552 | 6.7875445 | 4.03333 |

**A-2 CMSA A-2.1 Air**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 12 | 0.04 | 1006.4 | 1000 | 75.3538798779 | 0.64 |
| 2 | 15 | 0.05 | 1000 | 1000 | 73.8735572763 | 0 |
| 3 | 18 | 0.06 | 996.8 | 1000 | 72.3419511225 | 0.32 |
| 4 | 21 | 0.07 | 996.8 | 1000 | 70.7584094599 | 0.32 |
| 5 | 24 | 0.08 | 998.4 | 1000 | 69.1223453654 | 0.16 |
| 6 | 27 | 0.09 | 1001.6 | 1000 | 67.4332810759 | 0.16 |
| 7 | 30 | 0.1 | 1009.6 | 1000 | 65.6909061493 | 0.96 |

**A-6**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 6.00 | 0.04 | 2012.8 | 2000 | 37.9702708687 | 0.64 |
| 2 | 7.5 | 0.05 | 2001.6 | 2000 | 37.0808436634 | 0.08 |
| 3 | 9.00 | 0.06 | 1998.4 | 2000 | 36.1865748707 | 0.08 |
| 4 | 10.50 | 0.07 | 2000 | 2000 | 35.2883675836 | 0 |
| 5 | 12.00 | 0.08 | 2004.8 | 2000 | 34.3871856854 | 0.24 |
| 6 | 13.50 | 0.09 | 2009.6 | 2000 | 33.4840506095 | 0.48 |
| 7 | 15.00 | 0.1 | 2017.6 | 2000 | 32.5800372284 | 0.88 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 4.00 | 0.04 | 3043.2 | 3000 | 25.0457873392 | 1.44 |
| 2 | 5.00 | 0.05 | 3020.8 | 3000 | 24.4825770159 | 0.6933 |
| 3 | 6.00 | 0.06 | 3004.8 | 3000 | 23.9233603539 | 0.16 |
| 4 | 7.00 | 0.07 | 2995.2 | 3000 | 23.3685090423 | 0.16 |
| 5 | 8.00 | 0.08 | 2992 | 3000 | 22.8183901925 | 0.2667 |
| 6 | 9.00 | 0.09 | 2992 | 3000 | 22.2733649733 | 0.2667 |
| 7 | 10.00 | 0.1 | 2987.2 | 3000 | 21.7337872589 | 0.42667 |

**A-7**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 3.00 | 0.04 | 4152 | 4000 | 18.5378522462 | 3.8 |
| 2 | 3.75 | 0.05 | 4096 | 4000 | 18.1643929123 | 2.4 |
| 3 | 4.50 | 0.06 | 4052.8 | 4000 | 17.7952004927 | 1.32 |
| 4 | 5.25 | 0.07 | 4022.4 | 4000 | 17.4303889608 | 0.56 |
| 5 | 6.00 | 0.08 | 3992 | 4000 | 17.07006683 | 0.2 |
| 6 | 6.75 | 0.09 | 3968 | 4000 | 16.7143369565 | 0.8 |
| 7 | 7.50 | 0.1 | 3940.8 | 4000 | 16.3632963624 | 1.48 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 2.40 | 0.04 | 5145.6 | 5000 | 14.6504986457 | 2.912 |
| 2 | 3.00 | 0.05 | 5068.8 | 5000 | 14.3902789891 | 1.376 |
| 3 | 3.60 | 0.06 | 5014.4 | 5000 | 14.1333748359 | 0.288 |
| 4 | 4.20 | 0.07 | 4968 | 5000 | 13.8798148077 | 0.64 |
| 5 | 4.80 | 0.08 | 4928 | 5000 | 13.6296247878 | 1.44 |
| 6 | 5.62 | 0.09 | 4883.2 | 5000 | 13.2931900327 | 2.336 |
| 7 | 6.25 | 0.1 | 4840 | 5000 | 13.0390466996 | 3.2 |

**A-8**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h (mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 2.00 | 0.04 | 6202.4 | 6000 | 12.0890690512 | 3.3733 |
| 2 | 2.5 | 0.05 | 6152 | 6000 | 11.9002765354 | 2.5333 |
| 3 | 3.00 | 0.06 | 6054.4 | 6000 | 11.7139056428 | 0.90667 |
| 4 | 3.75 | 0.07 | 5950.4 | 6000 | 11.4388936904 | 0.82667 |
| 5 | 4.00 | 0.08 | 5921.6 | 6000 | 11.3484351419 | 1.30667 |
| 6 | 4.50 | 0.09 | 5864 | 6000 | 11.1693356925 | 2.26667 |
| 7 | 5.00 | 0.1 | 5795.2 | 6000 | 10.9926582245 | 3.4133 |

**A-2.2 Dielectric**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 12 | 0.04 | 1000.8 | 1000 | 1.1139896 | 68.762479 | 0.08 |
| 2 | 15 | 0.05 | 1002.4 | 1000 | 1.089159 | 67.089429 | 0.24 |
| 3 | 18 | 0.06 | 1004 | 1000 | 1.073211 | 65.450760 | 0.4 |
| 4 | 21 | 0.07 | 1002.4 | 1000 | 1.062103 | 63.834567 | 0.24 |
| 5 | 24 | 0.08 | 1004 | 1000 | 1.053922 | 62.235608 | 0.4 |
| 6 | 27 | 0.09 | 1008.8 | 1000 | 1.047645 | 60.651432 | 0.88 |
| 7 | 30 | 0.1 | 1011.2 | 1000 | 1.042677 | 59.080940 | 1.12 |

**A-9**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 6.00 | 0.04 | 2000.8 | 2000 | 1.25731 | 31.729942 | 0.04 |
| 2 | 7.5 | 0.05 | 2009.6 | 2000 | 1.195773 | 31.006941 | 0.48 |
| 3 | 9.00 | 0.06 | 2006.4 | 2000 | 1.157989 | 30.357052 | 0.32 |
| 4 | 10.50 | 0.07 | 2000 | 2000 | 1.13243 | 29.747789 | 0 |
| 5 | 12.00 | 0.08 | 1994.4 | 2000 | 1.113990 | 29.164841 | 0.28 |
| 6 | 13.50 | 0.09 | 1989.6 | 2000 | 1.100057 | 28.600945 | 0.52 |
| 7 | 15.00 | 0.1 | 1982.4 | 2000 | 1.089159 | 28.052050 | 0.88 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm)  predicted | %Error |
| 1 | 4.00 | 0.04 | 2993.6 | 3000 | 1.44295 | 19.605727 | 0.21333 |
| 2 | 5.00 | 0.05 | 3036.8 | 3000 | 1.32552 | 19.130148 | 1.01333 |
| 3 | 6.00 | 0.06 | 3046.4 | 3000 | 1.25731 | 18.756869 | 1.49333 |
| 4 | 7.00 | 0.07 | 3020.8 | 3000 | 1.21273 | 18.433877 | 0.69333 |
| 5 | 8.00 | 0.08 | 2988.8 | 3000 | 1.18132 | 18.140086 | 0.21333 |
| 6 | 9.00 | 0.09 | 2960 | 3000 | 1.157989 | 17.865288 | 0.90666 |
| 7 | 10.00 | 0.1 | 2926.4 | 3000 | 1.139979 | 17.603969 | 1.65333 |

**A-10**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) | %Error |
| 1 | 3.00 | 0.04 | 3952 | 4000 | 1.69291 | 13.762938 | 1.2 |
| 2 | 3.75 | 0.05 | 4065.6 | 4000 | 1.48686 | 13.359756 | 1.64 |
| 3 | 4.50 | 0.06 | 4084.8 | 4000 | 1.37527 | 13.091264 | 2.12 |
| 4 | 5.25 | 0.07 | 4064 | 4000 | 1.30529 | 12.881803 | 1.6 |
| 5 | 6.00 | 0.08 | 4030.4 | 4000 | 1.25731 | 12.703831 | 0.76 |
| 6 | 6.75 | 0.09 | 3988.8 | 4000 | 1.22236 | 12.544896 | 0.28 |
| 7 | 7.50 | 0.1 | 3947.2 | 4000 | 1.19577 | 12.398564 | 52.8 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) | %Error |
| 1 | 2.40 | 0.04 | 4830 | 5000 | 2.04762 | 10.563240 | 3.842 |
| 2 | 3.00 | 0.05 | 5096 | 5000 | 1.69291 | 10.149257 | 0.64 |
| 3 | 3.60 | 0.06 | 5128 | 5000 | 1.51765 | 9.9186117 | 1.824 |
| 4 | 4.20 | 0.07 | 5104 | 5000 | 1.41315 | 9.7583692 | 1.568 |
| 5 | 4.80 | 0.08 | 5027.2 | 5000 | 1.34375 | 9.6327249 | 0.64 |
| 6 | 5.62 | 0.09 | 4905.6 | 5000 | 1.27957 | 9.4912918 | 0.982 |
| 7 | 6.25 | 0.1 | 4808 | 5000 | 1.2445 | 9.3968568 | 2.272 |

**A-11**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 2.5 | 0.05 | 5880.8 | 6000 | 1.96527 | 8.239711 | 1.98666 |
| 2 | 3.00 | 0.06 | 6026.4 | 6000 | 1.69291 | 8.010813 | 0.44 |
| 3 | 3.75 | 0.07 | 6009.6 | 6000 | 1.48686 | 7.815923 | 0.16 |
| 4 | 4.00 | 0.08 | 5985.6 | 6000 | 1.44295 | 7.768709 | 0.24 |
| 5 | 4.50 | 0.09 | 5914.4 | 6000 | 1.3753 | 7.688583 | 1.42666 |
| 6 | 5.00 | 0.1 | 5828 | 6000 | 1.32552 | 7.621111 | 2.86667 |

**A-3 ETMSA A-3.1 Air**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 12 | 0.04 | 1000 | 1000 | 166.300714927 | 0 |
| 2 | 15 | 0.05 | 993.6 | 1000 | 161.779225742 | 0.64 |
| 3 | 18 | 0.06 | 993.6 | 1000 | 157.158332558 | 0.64 |
| 4 | 21 | 0.07 | 995.2 | 1000 | 152.427080495 | 0.48 |
| 5 | 24 | 0.08 | 995.2 | 1000 | 147.574641418 | 0.48 |
| 6 | 27 | 0.09 | 1000 | 1000 | 142.590688805 | 0 |
| 7 | 30 | 0.1 | 1006.4 | 1000 | 137.465857089 | 0.64 |

**A-12**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 6.00 | 0.04 | 2020.8 | 2000 | 81.6151452415 | 1.04 |
| 2 | 7.5 | 0.05 | 2000 | 2000 | 78.9791934068 | 0 |
| 3 | 9.00 | 0.06 | 2000 | 2000 | 76.3310872188 | 0 |
| 4 | 10.50 | 0.07 | 2003.2 | 2000 | 73.6743263718 | 0.10667 |
| 5 | 12.00 | 0.08 | 2014.4 | 2000 | 71.0127524199 | 0.72 |
| 6 | 13.50 | 0.09 | 2000 | 2000 | 68.3505398676 | 0 |
| 7 | 15.00 | 0.1 | 2004.8 | 2000 | 65.6921808655 | 0.24 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 4.00 | 0.04 | 3068 | 3000 | 53.1746257967 | 2.2667 |
| 2 | 5.00 | 0.05 | 3018 | 3000 | 51.5300433788 | 0.6 |
| 3 | 6.00 | 0.06 | 3000 | 3000 | 49.8999388382 | 0 |
| 4 | 7.00 | 0.07 | 2990 | 3000 | 48.2858989408 | 0.3333 |
| 5 | 8.00 | 0.08 | 2980 | 3000 | 46.6895083138 | 0.6667 |
| 6 | 9.00 | 0.09 | 2946 | 3000 | 45.1123409451 | 1.8 |
| 7 | 10.00 | 0.1 | 2932 | 3000 | 43.5559514079 | 2.26667 |

**A-13**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 3.00 | 0.04 | 4146 | 4000 | 39.0086455173 | 3.65 |
| 2 | 3.75 | 0.05 | 4048 | 4000 | 37.914033819 | 1.2 |
| 3 | 4.50 | 0.06 | 4004 | 4000 | 36.8348971144 | 0.1 |
| 4 | 5.25 | 0.07 | 3966 | 4000 | 35.7717750465 | 0.85 |
| 5 | 6.00 | 0.08 | 3934 | 4000 | 34.7251831803 | 1.65 |
| 6 | 6.75 | 0.09 | 3978 | 4000 | 33.6956111918 | 0.55 |
| 7 | 7.50 | 0.1 | 3822 | 4000 | 32.6835211639 | 4.45 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 2.40 | 0.04 | 5283.2 | 5000 | 30.3605704629 | 5.664 |
| 2 | 3.00 | 0.05 | 5127.2 | 5000 | 29.5955944487 | 2.544 |
| 3 | 3.60 | 0.06 | 5040.8 | 5000 | 28.8430835912 | 0.816 |
| 4 | 4.20 | 0.07 | 4966.4 | 5000 | 28.1031867042 | 0.672 |
| 5 | 4.80 | 0.08 | 4901.6 | 5000 | 27.3760378747 | 1.968 |
| 6 | 5.62 | 0.09 | 4762.4 | 5000 | 26.4030980294 | 4.752 |
| 7 | 6.25 | 0.1 | 4697.6 | 5000 | 25.6720834424 | 6.048 |

**A-14**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | A predicted | %Error |
| 1 | 2.00 | 0.04 | 6523.2 | 6000 | 24.4388347769 | 8.72 |
| 2 | 2.5 | 0.05 | 6280.8 | 6000 | 23.8868820781 | 4.68 |
| 3 | 3.00 | 0.06 | 6144 | 6000 | 23.3443295343 | 2.4 |
| 4 | 3.75 | 0.07 | 5976 | 6000 | 22.5481451217 | 0.4 |
| 5 | 4.00 | 0.08 | 5924.8 | 6000 | 22.2874562197 | 1.2533 |
| 6 | 4.50 | 0.09 | 5824 | 6000 | 21.7731329745 | 2.9333 |
| 7 | 5.00 | 0.1 | 5665.6 | 6000 | 21.2682051469 | 5.57333 |

**A-3.3 Dielectric**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 12 | 0.04 | 1005.6 | 1000 | 1.1139896 | 147.61658 | 0.56 |
| 2 | 15 | 0.05 | 1004 | 1000 | 1.089159 | 143.94687 | 0.4 |
| 3 | 18 | 0.06 | 1004 | 1000 | 1.073211 | 140.24326 | 0.4 |
| 4 | 21 | 0.07 | 1003.2 | 1000 | 1.062103 | 136.53055 | 0.32 |
| 5 | 24 | 0.08 | 1005.6 | 1000 | 1.053922 | 132.81946 | 0.56 |
| 6 | 27 | 0.09 | 1008.8 | 1000 | 1.047645 | 129.11490 | 0.88 |
| 7 | 30 | 0.1 | 1012 | 1000 | 1.042677 | 125.41917 | 1.2 |

**A-15**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 6.00 | 0.04 | 2001.6 | 2000 | 1.25731 | 67.886050 | 0.08 |
| 2 | 7.5 | 0.05 | 2008 | 2000 | 1.195773 | 66.173384 | 0.4 |
| 3 | 9.00 | 0.06 | 2005.6 | 2000 | 1.157989 | 64.517390 | 0.28 |
| 4 | 10.50 | 0.07 | 2001.6 | 2000 | 1.13243 | 62.899630 | 0.08 |
| 5 | 12.00 | 0.08 | 2000 | 2000 | 1.113990 | 61.312215 | 0 |
| 6 | 13.50 | 0.09 | 1993.6 | 2000 | 1.100057 | 59.751416 | 0.32 |
| 7 | 15.00 | 0.1 | 1991.2 | 2000 | 1.089159 | 58.215418 | 0.44 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 4.00 | 0.04 | 3000 | 3000 | 1.44295 | 41.891462 | 0 |
| 2 | 5.00 | 0.05 | 3019.2 | 3000 | 1.32552 | 40.838128 | 0.64 |
| 3 | 6.00 | 0.06 | 3016.8 | 3000 | 1.25731 | 39.901921 | 0.56 |
| 4 | 7.00 | 0.07 | 3003.2 | 3000 | 1.21273 | 39.028503 | 0.10666 |
| 5 | 8.00 | 0.08 | 2982.4 | 3000 | 1.18132 | 38.195485 | 0.58666 |
| 6 | 9.00 | 0.09 | 2964.9 | 3000 | 1.157989 | 37.391928 | 1.17 |
| 7 | 10.00 | 0.1 | 2948 | 3000 | 1.139979 | 36.611894 | 1.7333 |

**A-16**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 3.00 | 0.04 | 3947.2 | 4000 | 1.69291 | 29.643877 | 1.32 |
| 2 | 3.75 | 0.05 | 4022.4 | 4000 | 1.48686 | 28.823072 | 0.56 |
| 3 | 4.50 | 0.06 | 4020.8 | 4000 | 1.37527 | 28.182732 | 0.52 |
| 4 | 5.25 | 0.07 | 3996.8 | 4000 | 1.30529 | 27.625443 | 0.08 |
| 5 | 6.00 | 0.08 | 3963.2 | 4000 | 1.25731 | 27.115021 | 0.92 |
| 6 | 6.75 | 0.09 | 3921.6 | 4000 | 1.22236 | 26.634903 | 1.96 |
| 7 | 7.50 | 0.1 | 3876.8 | 4000 | 1.19577 | 26.176450 | 3.08 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 2.40 | 0.04 | 4827.2 | 5000 | 2.04762 | 22.693580 | 3.456 |
| 2 | 3.00 | 0.05 | 5000 | 5000 | 1.69291 | 21.899140 | 0 |
| 3 | 3.60 | 0.06 | 5025.6 | 5000 | 1.51765 | 21.380399 | 0.512 |
| 4 | 4.20 | 0.07 | 5000 | 5000 | 1.41315 | 20.970988 | 0 |
| 5 | 4.80 | 0.08 | 4944 | 5000 | 1.34375 | 20.617065 | 1.12 |
| 6 | 5.62 | 0.09 | 4857.6 | 5000 | 1.27957 | 20.184065 | 2.848 |
| 7 | 6.25 | 0.1 | 4780.8 | 5000 | 1.2445 | 19.876560 | 4.384 |

**A-17**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No. | h+Δ(mm) | h/λ0 | fie3d(Mhz) | fANN(Mhz) | E(ref) | a(mm) predicted | %Error |
| 1 | 2.00 | 0.04 | 5536 | 6000 | 2.59036 | 18.395785 | 7.7333 |
| 2 | 2.5 | 0.05 | 5960 | 6000 | 1.96527 | 17.462394 | 0.66667 |
| 3 | 3.00 | 0.06 | 6040 | 6000 | 1.69291 | 16.975917 | 0.66667 |
| 4 | 3.75 | 0.07 | 5984 | 6000 | 1.48686 | 16.495218 | 0.26667 |
| 5 | 4.00 | 0.08 | 5940.8 | 6000 | 1.44295 | 16.365316 | 0.98667 |
| 6 | 4.50 | 0.09 | 5859.2 | 6000 | 1.3753 | 16.130512 | 2.34667 |
| 7 | 5.00 | 0.1 | 5761.6 | 6000 | 1.32552 | 15.918342 | 3.9733 |

**A-18**

**REFERENCES**

1)Global Journal for researches in Electrical and Electronics Engineering -Artificial

Neural Network to design Circular Microstrip Antenna - Sami Bedra

2) Deshmukh, Amit A., and Ray, K. P., Analysis of shorted Plate Compact and Broadband

Microstrip Antenna, IEEE Magazine on Antennas and Propagation, vol. 55, no. 6, December 2013, pp. 100 – 113.

3)Deshmukh, Amit A., Kulkarni, Shivali D., Venkata, A. P. C., and Phatak, Neelam V., Artificial Neural Network Model for Suspended Rectangular Microstrip Antennas, Proceedings of ICAC3 - 2015, 1st &amp; 2nd April 2014, Mumbai, India

4)S ¸. Sagıroglu, K. Guney, M. Erler, “Resonant frequency calculation for circular microstrip antennas using artificial neural networks”, International Journal of RF and Microwave Computer-Aided Engineering,Vol. 8, No. 3, pp. 270- 277, 1998.

5)D. Karaboga, K. Guney, S ¸ . Sagıroglu, M. Erler, “Neural computation of resonant frequency of electrically thin and thick rectangular microstrip antennas”, Microwaves, Antennas and Propagation, IEE Proceedings-Vol. 146, No. 2, pp. 155 - 159, April 1999

6)IE3D 12, Zeland Software, Freemont, USA

7)Kushwah, V. S., Tomar, G. S., 2009. Design of Microstrip Patch Antennas Using Neural Network, Proceedings of Third Asia International Conference on Modelling &amp; Simulation, Bali, (DOI - 10.1109/AMS.2009.12).

8) C.A. Balanis .

9)Design of Circular Microstrip Antenna Using Artificial Neural Networks-

Jagtar Singh, A.P. Singh, and T.S. Kamal, Member, IAENG.