

7-3-2012

# Neural network field programmable gate array (FPGA) controllers for reconfigurable antennas

Eyad Al Zuraiqi

Follow this and additional works at: [https://digitalrepository.unm.edu/ece\\_etds](https://digitalrepository.unm.edu/ece_etds)

---

## Recommended Citation

Al Zuraiqi, Eyad. "Neural network field programmable gate array (FPGA) controllers for reconfigurable antennas." (2012).  
[https://digitalrepository.unm.edu/ece\\_etds/8](https://digitalrepository.unm.edu/ece_etds/8)

This Dissertation is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact [disc@unm.edu](mailto:disc@unm.edu).

**Eyad Talal Al Zuraiqi**

*Candidate*

**Electrical & Computer Engineering Department**

*Department*

This dissertation is approved, and it is acceptable in quality and form for publication on microfilm.

*Approved by the Dissertation Committee:*

Dr. Christos Christodoulou,  
*Chairperson*

---

Dr. Howard Pollard

---

Dr. Jamesina Simpson

---

Dr. Mahmoud Taha

---

---

---

---

---

Accepted:

---

*Dean, Graduate School*

---

*Date*

# **Neural Network Field Programmable Gate Array (FPGA) Controllers for Reconfigurable Antennas**

by

**Eyad Talal Al Zuraiqi**

B.S., Electrical Engineering, Mu'tah University, 2003

MSc, Communications and Electronics Engineering, Jordan University of  
Science and Technology, 2006

DISSERTATION

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Doctor of Philosophy  
Engineering

The University of New Mexico

Albuquerque, New Mexico

April, 2012

©2012, Eyad Talal Al Zuraiqi

## **Dedication**

*To my parents, for their support, love and patience.*

## **Acknowledgments**

First of all, I would like to express my sincere thanks and gratitude towards Prof. Christos Christodoulou, my advisor and dissertation chair, for continuous encourage, guidance and support through beautiful years of academic coursework, writing and doing experiments of this dissertation. Prof. Christos patiently went over the drafts suggesting and numerous improvements.

Thanks go to other committee members, Dr. Howard Pollard, Dr .Jamesina Simpson, Dr. Mahmoud Taha, for their valuable recommendations pertaining to this study and assistance in my professional development.

Deep gratitude goes to my beloved family and friends, for their constant encouragement, support, love and patience they have bestowed upon me. Your encouragement is greatly appreciated.

# **Neural Network Field Programmable Gate Array (FPGA) Controllers for Reconfigurable Antennas**

by

**Eyad Talal Al Zuraiqi**

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Doctor of Philosophy  
Engineering

The University of New Mexico

Albuquerque, New Mexico

April, 2012

# **Neural Network Field Programmable Gate Array (FPGA) Controllers for Reconfigurable Antennas**

**By**

**Eyad Talal Al Zuraiqi**

B.S., Electrical Engineering, Mu'tah University, 2003

MSc, Communication and Electronics Engineering, Jordan University for Science and Technology, 2006

## **Abstract**

Advantages of reconfigurable antennas are numerous, but limited by the method of controlling their configuration. This dissertation proposes to utilize the advantages of both Neural Networks (NN) and Field Programmable Gate Arrays (FPGAs) to overcome this dilemma.

In this work, the methodology of modeling of reconfigurable antennas using neural network embedded on an FPGA board is presented. This work shows a new approach of modeling reconfigurable antennas using neural networks in Matlab, a code is written to generate a NN for any antenna (or any reconfigurable system in general) by providing input/output data of the antenna.

An HDL code is generated using Xilinx System Generator and sent to an FPGA board using Xilinx ISE. With a NN embedded on the FPGA board, we have a highly



reconfigurable system in real time controller that thinks exactly as the system it models. This “brain” is connected to the antenna and becomes the decision maker in antenna switching or reconfiguration. Also, with the new approach of using Matlab to generate HDL code; this work opens the door to those who are interested in implementing designs on FPGAs without having enough knowledge in HDL programming.

Different types of reconfigurable antennas with different way of reconfigurability are modeled and simulated. NN models show great match with measured antennas data. NN\_FPGA controller is built for each antenna.

## Table of Contents

List of Figures .....	xi
List of Tables .....	xiv
Chapter I: Introduction .....	1
1.1 Literature Review .....	1
1.2 Dissertation Contributions .....	2
Chapter II: Theoretical Background.....	4
2.1 Basic Review of Artificial Neural Networks:.....	4
2.1.1 Concept of NN .....	4
2.1.2 Neural Network Learning .....	6
2.1.3 Connections Weights.....	6
2.1.4 Back-propagation .....	7
2.1.5 Mathematical model of a neuron .....	7
2.1.6 activation functions .....	10
2.2 Introduction to Reconfigurable Antennas.....	13
2.2.1 Frequency Response.....	14
2.2.2 Frequency Response Reconfigurability.....	15
2.2.3 Microstrip Patch Antennas .....	18
2.3 Field Programmable Gate Arrays .....	23

2.3.1	JTAG 1149.1 Standard .....	24
2.3.2	TAP Controller.....	25
2.3.3	The Instruction Register .....	27
Chapter III: NN-FPGA Controller Design .....		30
3.1	NN Implementation.....	30
3.2	NN modeling on FPGA procedure.....	31
Chapter IV: Examples, Simulations and Results .....		41
4.1	Example 1: Reconfigurable Filtenna Antenna .....	41
4.2	Example 2: Microstrip Patch Antenna .....	50
4.3	Example 3: Rotatable Microstrip Antenna .....	57
4.4	Example 4: Planar Ultra-Wideband Antenna .....	61
4.5	Example 5: Reconfigurable filter .....	65
4.6	Example 6: Star Antenna .....	69
Chapter V: Conclusions and future work .....		74
Appendix A: Matlab Codes .....		76
Appendix B: Input/Output Pairs .....		94
Appendix C: VHDL code .....		118
Appendix D: Simulation models .....		133
References.....		144

## LIST OF FIGURES

Fig. 2.1: Biological Neural connections .....	4
Fig. 2.2: Artificial neural network topology .....	4
Fig. 2.3: Neuron mode .....	7
Fig. 2.4: Linear activation function .....	10
Fig. 2.5: Sigmoid activation function, with $a = 1$ .....	11
Fig. 2.6: Structure of a conventional rectangular MSA .....	18
Fig. 2.7: Microstrip feeding and fringing field lines .....	20
Fig. 2.8: Microstrip fringing field lines .....	21
Fig. 2.9: Architecture of the TAP controller .....	25
Fig. 3.1: NN modeling on FPGA procedure .....	30
Fig. 3.2: Sigmoid function segments .....	33
Fig. 3.3: Sigmoid function Segment 1 .....	33
Fig. 3.4: Sigmoid function Segment 2 .....	34
Fig. 3.5: Sigmoid function Segment 3 function .....	34
Fig. 3.6: Sigmoid function Segment 4 function .....	35
Fig. 3.7: Sigmoid function Segment 5 .....	35
Fig. 3.8: Sigmoid Simulink model .....	36
Fig. 3.9: System Generator block .....	37
Fig. 3.10: FPGA board connections .....	38
Fig. 4.1: Filtenna antenna structure .....	40
Fig. 4.2: Filtenna antenna NN model .....	41

Fig. 4.3: Simulink NN_FPGA model of the antenna .....	43
Fig. 4.4: Hidden layer of filtenna antenna .....	44
Fig. 4.5: Output layer of filtenna antenna .....	45
Fig. 4.6: Output neuron structure .....	46
Fig. 4.7: NN output vs measured antenna response .....	47
Fig. 4.8: Antenna Structure and dimensions .....	49
Fig. 4.9: Neural network representation for this antenna .....	50
Fig. 4.10: NN training error .....	51
Fig. 4.11: NN output (red) vs. measured antenna response (blue) .....	52
Fig. 4.12: NN output vs measured antenna response for SW1 off and SW2 off ...	53
Fig. 4.13: NN output vs measured antenna response for SW1 on and SW2 on ...	53
Fig. 4.14: NN output vs measured antenna response for SW1 off and SW2 on ...	54
Fig. 4.15: NN output vs measured antenna response for SW1 on and SW2 off ...	54
Fig. 4.16: Antenna Structure .....	56
Fig. 4.17: Neural network representation for this antenna .....	56
Fig. 4.18 : NN output vs measured antenna response for shape 1 .....	57
Fig. 4.19 : NN output vs measured antenna response for shape 2 .....	58
Fig. 4.20 : NN output vs measured antenna response for shape 3 .....	58
Fig. 4.21: Antenna Structure and dimensions .....	60
Fig. 4.22: NN output vs measured antenna response for case 0100 .....	61
Fig. 4.23: NN output vs measured antenna response for case 1000 .....	62
Fig. 4.24: NN output vs measured antenna response for case 1111 .....	62
Fig. 4.25: NN output vs measured antenna response for case 0000 .....	63

Fig. 4.26: Reconfigurable filter .....	64
Fig. 4.27: NN output vs measured antenna response for Model 1 .....	66
Fig. 4.28: NN output vs measured antenna response for Model 2 .....	66
Fig. 4.29: NN output vs measured antenna response for Model 3 .....	67
Fig. 4.30: Star antenna structure .....	68
Fig. 4.31: NN model for Star antenna .....	69
Fig. 4.32: Determining number of hidden neurons .....	70
Fig. 4.33: NN output vs measured antenna response for case 111000 .....	71
Fig. 4.34: NN output vs measured antenna response for case 110000 .....	71

## LIST OF TABLES

Table 2.1: Test Access Port Connections .....	24
Table 4.1: Measured S11 for filtenna antenna .....	41
Table 4.2: Neural network parameters for Filtenna antenna .....	42
Table 4.3: Filtenna NN simulation .....	47
Table 4.4: Neural network parameters for microstrip antenna .....	49
Table 4.5: NN parameters of Rotatable Microstrip Antenna .....	56
Table 4.6: NN parameters of Planar Ultra-Wideband Antenna .....	60
Table 4.7: NN parameters of Broadband Tapered Slot Antenna .....	64
Table 4.8: Star antenna neural network parameters .....	69

# CHAPTER I: INTRODUCTION

## 1.1 Literature Review

Neural networks (NN) were initially studied by computer and cognitive scientists in the late 1950s and early 1960s in an attempt to model sensory perception in biological organisms [1]. Neural networks have been applied to many problems since they were first introduced, including pattern recognition, handwritten character recognition, speech recognition, financial and economic modeling, and next-generation computing models.

Neural networks have emerged in recent years as a powerful technique for modeling general input/output relationships. The distinguished characteristics of Artificial Neural Networks such as to learn from data, to generalize patterns in data and to model nonlinear relationships makes them a good candidate for many different fields of engineering.

In addition, theoretical and numerical research was conducted in using Artificial Neural Networks (ANN) in electromagnetics [2-8]. Neural Networks, due to their high speed computational capability, can be employed with reconfigurable antennas to achieve reconfigurability in real time, and to avoid the computational complexities involved in analyzing reconfigurable antennas. NNs have been applied in microwave engineering and applied in antenna and array analysis [9,10] with encouraging results.

However, a fully reconfigurable antenna implementation requires a considerable increase in processing requirements. So, by using FPGAs (Field Programmable Gate



Arrays), powerful DSP devices are used to handling the high performance requirements at sampled data rates [11-13]. Furthermore, we can take advantage of the FPGA's flexibility for directly handling acquisition control and other DSP functions, such as digital down-conversion, demodulation, and match filtering.

## 1.2 Dissertation Contributions

This work proposes to utilize the advantages of both Neural Networks (NN) and Field Programmable Gate Arrays (FPGAs) to overcome this problem.

Neural Networks, due to their high speed computational capability, can be employed with reconfigurable antennas to achieve reconfigurability in real time, and to avoid the computational complexities involved in analyzing reconfigurable antennas.

Neural Networks models rely on massive parallel computation. Thus, the high speed operation in real time applications can be achieved only if the NNs are implemented using parallel hardware architecture. Neural Networks modeling on FPGA preserves the parallel architecture of the neurons within layers and allows flexibility in reconfiguration.

The dissertation shows a new approach of modeling reconfigurable antennas using neural networks in Matlab, a code is written to generate a NN for any antenna (or any reconfigurable system in general) by providing input/output data of the antenna. In order for the generated NN to be implemented on FPGA board, it need to be built using

“logic” blocks. Chapter 3 shows how the NN is rebuilt using Xilinx System Generator blocks only.

An HDL code is generated using Xilinx System Generator and sent to an FPGA board using Xilinx ISE. With a NN embedded on the FPGA board, we have a highly reconfigurable system in real time controller that thinks exactly as the system it models. This “brain” is connected to the antenna and becomes the decision maker in antenna switching or reconfiguration.

Also, with the new approach of using Matlab to generate HDL code, this work opens the door to those who are interested in implementing designs on FPGAs without having enough knowledge in HDL programming.

## CHAPTER II: THEORETICAL BACKGROUND

This chapter gives the reader a basic theoretical background to main components of the dissertation; neural network, reconfigurable antennas, and FPGAs.

### 2.1 Basic Review of Artificial Neural Networks:

Neural Network (NN) is a highly interconnected network of information-processing elements that mimics the connectivity and functioning of the human brain. Neural networks provide some insight into the way the human brain works. One of the most significant strengths of neural networks is their ability to learn from a limited set of examples.

#### 2.1.1 Concept of NN

Neural networks are subdivided into two categories, artificial neural networks and biological neural networks. Artificial neural networks are used to simulate the structure and functioning of biological neural networks. The most familiar biological neural network is the human brain.

The basic element in a NN is the neuron. A biological neuron (as shown in figure (2.1)) is a nerve cell with all of its processes. Neurons have three main parts [14]. A

central cell body, called the soma, and two different types of branched, treelike structures that extend from the soma, called dendrites, and axons. In a form of electrical impulses, information from other neurons enters the dendrites at connection points called synapses. The information flows from the dendrites to the soma, where it is processed. The output signal is then sent down the axon to the synapses of other neurons.

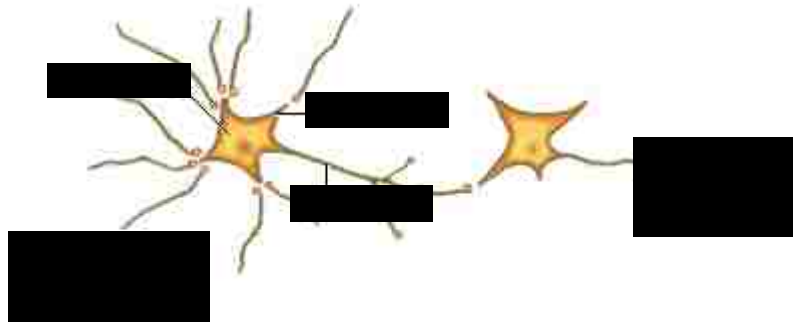


Fig. 2.1. Biological Neural connections [14]

Artificial neurons (figure (2.2)) are designed to mimic the function of biological neurons.

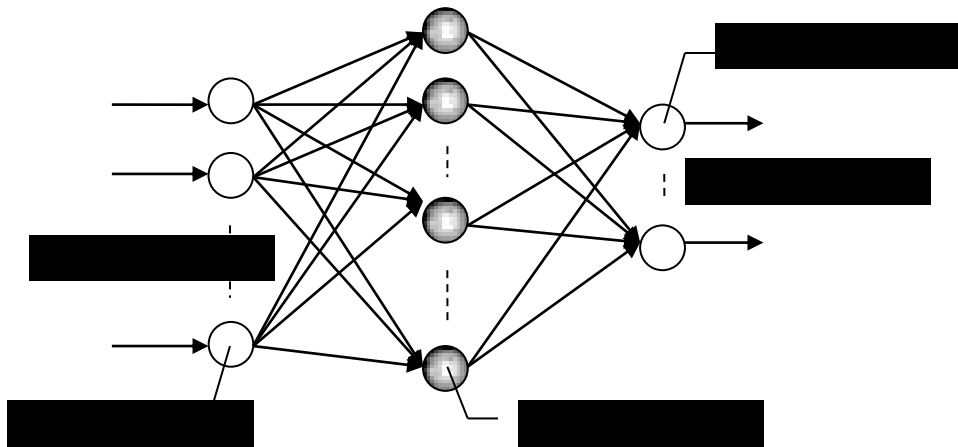


Fig. 2.2. Artificial neural network topology

Artificial neurons have structures that designed to mimic the function of biological neurons. The main body of an artificial neuron is called a node or unit. Artificial neurons may be physically connected to one another by wires that mimic the connections between biological neurons, if, for instance, the neurons are simple integrated circuits. However, neural networks are usually simulated on traditional computers, in which case the connections between processing nodes are not physical but instead they are virtual connections.

Neural network architecture has three layers. The first layer is called the input layers and is the only layer exposed to external signals. The input layer transmits signals to the neurons in the next layer, which is called a hidden layer. The hidden layer extracts relevant features or patterns from the received signals. Those features or patterns that are considered important are then directed to the final layer of the network, the output layer.

### 2.1.2 Neural Network Learning

The ‘strength’ of the connections between relevant neurons determines the ‘strength’ of the memory. Important information that needs to be remembered may cause the brain to constantly reinforce the pathways between the neurons that form the memory, while relatively unimportant information will not receive the same degree of reinforcement.

### 2.1.3 Connections Weights

To simulate the way in which biological neurons reinforce certain axon-dendrite pathways, the connections between artificial neurons in a neural network are given

adjustable connection weights. When signals are received and processed by a node, they are multiplied by a weight, and then added up.

#### 2.1.4 Back-propagation

The most widely used scheme for adjusting the connection weights is called error back-propagation. The back-propagation learning scheme compares a neural network's output to a target output and calculates an error adjustment for each of the nodes in the network. The neural network adjusts the connection weights according to the error values assigned to each node, starting with the connections between the last hidden layer and the output layer. After the network has made adjustments to this set of connections, it calculates error values for the next previous layer and makes adjustments. The back-propagation algorithm continues in this way, adjusting all of the connection weights between the hidden layers until it reaches the input layer. At this point it is ready to calculate another output.

#### 2.1.5 Mathematical model of a neuron

Neuron is the information processing unit in a neural network. The mathematical part is extracted from Haykin's 'Neural Networks' book [15]. Figure (2.3) shows the model of a neuron, which represents the basis for building neural networks.

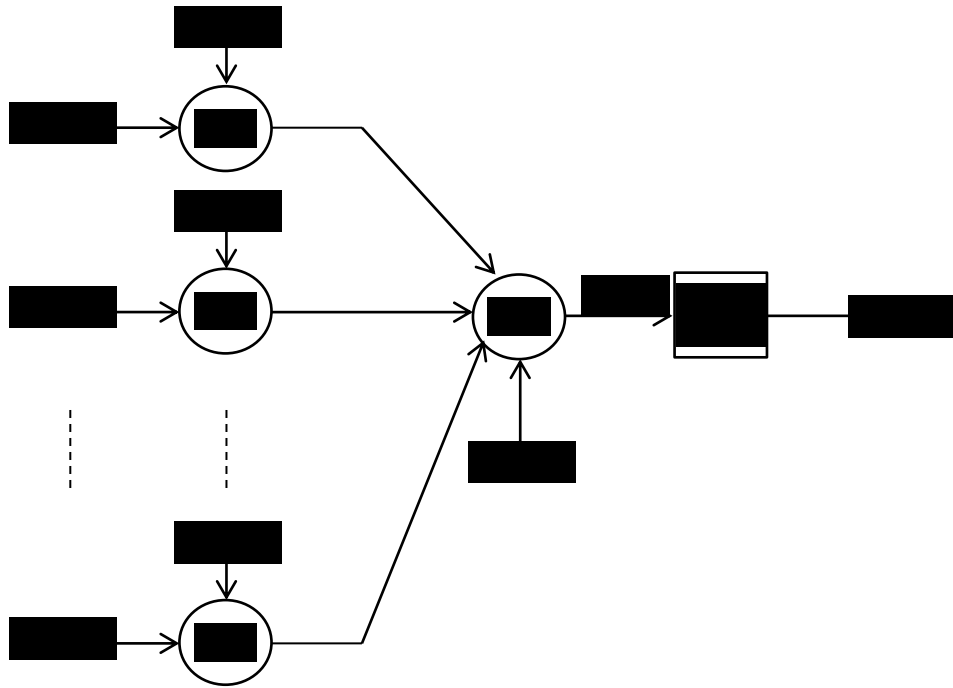


Fig. 2.3. Neuron model

As shown in Figure 2.3, a neuron has three main components:

1. Synaptic links, which is characterized by the synaptic weights ( $w$ ). Each input signal ( $x_i$ ) is multiplied by the synaptic weight ( $w_{ki}$ ).
2. Adder, for adding up the input signals, weighted by synaptic weights.
3. Activation function, for limiting the amplitude of neurons output to some finite limit.

A neuron  $k$  can be mathematically described using the following pair of equations,

$$v_k = \sum_{j=1}^m w_{kj} x_j \quad 2.1$$

and

$$y_k = \varphi(v_k + b_k) \quad 2.2$$

Where  $x_j, j = 1, 2, \dots, m$  are the input signals,  $w_{kj}, j = 1, 2, \dots, m$  are the synaptic weights of neuron  $k$ ,  $u_k$  is the linear combiner output due to the input signal,  $b_k$  is the bias,  $\varphi(\cdot)$  is the activation function.

Neurons in each layer compute the function signal that appears on their output and calculate an estimate of the gradient vector needed for the backward pass of the signal that determines the change in the weights of each neuron.

The function of an output neuron can be described using equations (2.3-5). The instantaneous error at the neuron's output can be written:

$$e_i(n) = d_i(n) - y_i(n) \quad 2.3$$

where  $d_i(n)$  is the desired output for the neuron  $i$ , and  $y_i(n)$  is the output function signal of neuron  $i$  at the  $n$ th iteration. The total instantaneous error of all neurons is then:

$$E(n) = \frac{1}{2} \sum_{i=1}^I e_i^2(n) \quad 2.4$$

where  $I$  is the number of neurons in the network's output layer. If  $N$  is the total number of examples contained in the training set, then the average squared error energy is:

$$E_{ave} = \frac{1}{N} \sum_{n=1}^N E(n) \quad 2.5$$



The purpose of the neural network is to change the weights of its neurons in such a way that the total network's error will be minimized; the new weights are expressed as:

$$w_{ij}(n + 1) = w_{ij}(n) + \Delta w_{ij}(n) \quad 2.6$$

Where,

$$\Delta w_{ij}(n) = -\mu \frac{\partial E(n)}{\partial w_{ij}(n)} \quad 2.7$$

and  $\mu$  is the learning rate parameter of the backpropagation algorithm.

The learning rate is a parameter that controls how small or large are the changes to the synaptic weights in the network from one iteration to the next. The improvement in the learning is attained at the cost of a slower rate of learning. A too large  $\mu$  may speed up the rate of learning, but the resulting changes in the synaptic weights assume such a form that the network may become unstable.

#### 2.1.6 activation functions

The activation function defines the output of a neuron in terms of the induced local field  $v$ . Through this dissertation, we will use two types of activation functions:

1. Step activation function, for this type of activation function, shown in figure (2.4),

we have:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad 2.8$$

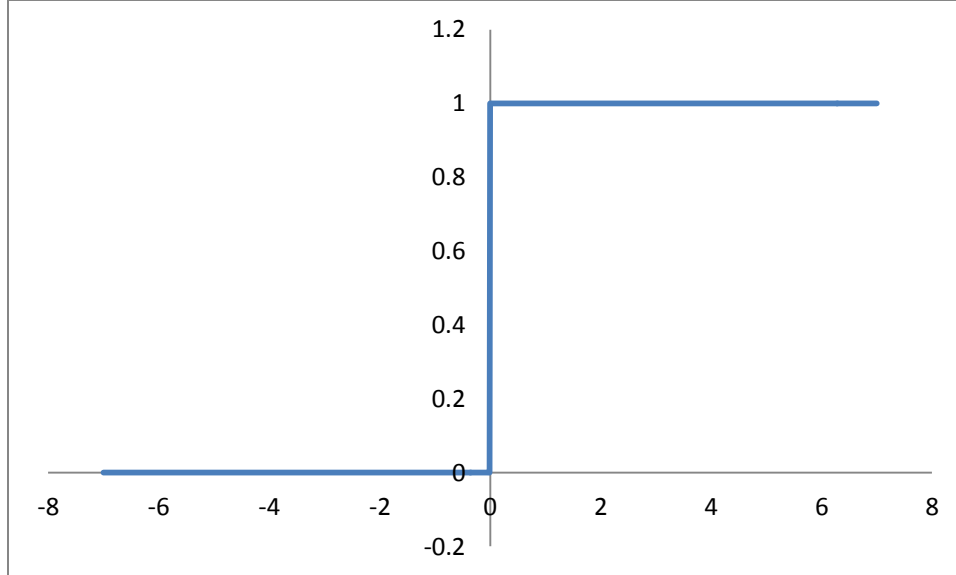


Fig. 2.4: Step activation function

Correspondingly, the output of a neuron  $k$  using such an activation function is expressed as

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases} \quad 2.9$$

Where  $v_k$  is the induced local field of the neuron, that is,

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad 2.10$$

2. Sigmoid function, the sigmoid function can be defined using the following equation,

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad 2.11$$

Where “a” is the slope parameter of the sigmoid function. By changing the parameter “a”, we can obtain a sigmoid function of different slopes, as shown in figure (2.5).

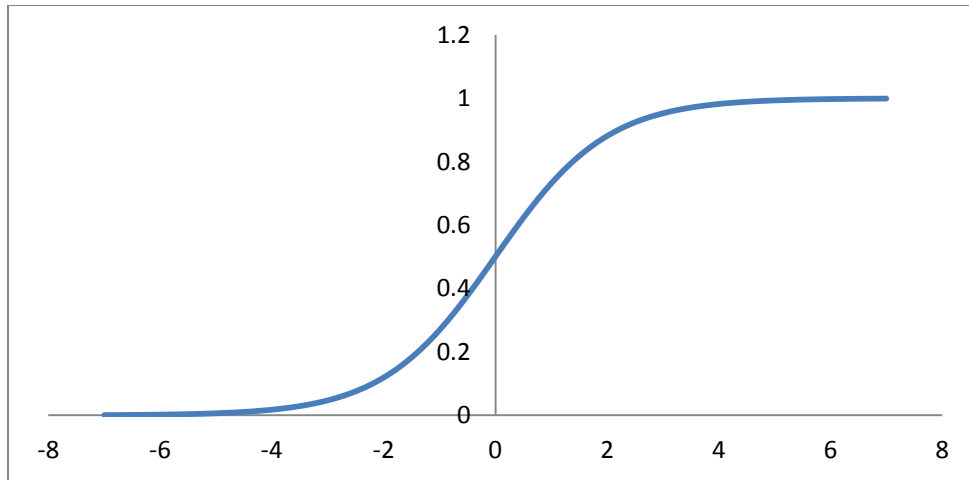


Fig. 2.5: Sigmoid activation function, with  $a = 1$

## 2.2 Introduction to Reconfigurable Antennas

This section provides introduction to definitions and the operation of reconfigurable antennas in general [16-17]. Specific reconfigurable antennas used in this dissertation will be explained individually in the simulations and results chapter later.

Antennas are critical components of communication and radar systems. Different types of antennas have developed during the past 50 years in both wireless communication and radar systems. These varieties include dipoles/monopoles, loop antennas, slot/horn antennas, reflector antennas, microstrip antennas, log periodic antennas, helical antennas, dielectric/lens antennas, and frequency-independent antennas. Each category possesses inherent benefits and detriments that make them more or less suitable for particular applications. When faced with a new system design, engineers change and adapt these basic antennas, using theoretical knowledge and general design guidelines as starting points to develop new structures that often produce acceptable results.

Nevertheless, the choice of an antenna from the families mentioned above also imposes restrictions on the overall system performance that arises because the antenna characteristics are fixed.

Reconfigurability is the capacity to change an individual radiator's fundamental operating characteristics through electrical, mechanical, or other means. Thus, under this definition, the traditional phasing of signals between elements in an array to achieve beam forming and beam steering does not make the antenna "reconfigurable" because the antenna's basic operating characteristics remain unchanged in this case.

Reconfigurable antennas should be able to alter their operating frequencies, impedance bandwidths, polarizations, and radiation patterns independently to accommodate changing operating requirements. However, the development of these antennas poses significant challenges to both antenna and system designers. These challenges lie not only in obtaining the desired levels of antenna functionality but also in integrating this functionality into complete systems to arrive at efficient and cost-effective solutions. As in many cases of technology development, most of the system cost will come not from the antenna but the surrounding technologies that enable reconfigurability.

Traditional characterization of any antenna requires two types of information: the input impedance characteristic over frequency (typically called the frequency response) and the radiation characteristic (or radiation pattern). Usually, frequency response is considered first because without a reasonable input impedance match, a transmitting system may suffer from severe reflections that could damage other components and waste power, whereas receiving systems will suffer from reduced sensitivity and require additional signal amplification. Once an antenna's frequency response is known, the radiation patterns are examined. This chapter briefly reviews both the frequency and radiation characteristics of antennas that can be manipulated through reconfiguration of physical and material parameters as will be shown later.

### 2.2.1 Frequency Response

The frequency response of an antenna is defined as its input impedance over frequency. Complex input impedance (in  $Z_{in}(\omega) = R(\omega) + jX(\omega)$  form, where  $\omega = 2\pi f$  is the radian frequency) provides the ability to consider the antenna as a circuit element. As

such, the antenna's input impedance can then be used to determine the reflection coefficient ( $\Gamma$ ) and related parameters, such as voltage standing wave ratio (VSWR) and return loss (RL), as a function of frequency are given as

$$\Gamma = \frac{Z_{in}(\omega) - Z_0}{Z_{in}(\omega) + Z_0} \quad (2.12)$$

$$VSWR = \frac{V_{max}}{V_{min}} = \frac{1 + |\Gamma|}{1 - |\Gamma|} \quad (2.13)$$

$$RL = -20 \log |\Gamma| \quad (2.14)$$

The efficiency of a traditional antenna is typically determined by the ohmic losses created by imperfect conductors and/or dielectrics.

In reconfigurable antennas, losses may be increased, or other sources of loss may arise. These include losses incurred by any solid-state devices (such as diodes, field-effect transistors (FETs), plasma-based devices, etc.) or other switches or materials used to enable reconfiguration. Reconfigurable antennas based on controlled changes in dielectric or magnetic properties of materials (such as ferroelectrics and ferrites) usually experience more loss because of the presence of often-substantial electrical conductivities of these materials.

Effective losses may also be caused by current leakage through control lines or undesired radiation by circuitry used to enable the antenna's reconfiguration.

### 2.2.2 Frequency Response Reconfigurability

Frequency-reconfigurable antennas (also called tunable antennas) can be classified into two categories: continuous and switched. Continuous frequency-tunable antennas allow

for smooth transitions within or between operating bands without jumps. Switched tunable antennas, on the other hand, use some kind of switching mechanism to operate at distinct and/or separated frequency bands.

Both kinds of antennas in general share a common theory of operation and reconfiguration; the main differences are in the extent of the effective length changes that enable operation over different frequency bands and the devices and/or means used to achieve these changes.

Many common antennas, including linear antennas, loop antennas, slot antennas, and microstrip antennas, are usually operated in resonance. In these cases, the effective electrical length of the antenna largely determines the operating frequency, its associated bandwidth (typically no more than about 10% and usually around 1% to 3% for a single resonance), and the current distribution on the antenna that dictates its radiation pattern. For instance, for a traditional linear dipole antenna, the first resonance occurs at a frequency where the antenna is approximately a half wavelength long, and the resulting current distribution results in an omnidirectional radiation pattern centered on and normal to the antenna axis. In this case, if one wants the antenna to operate at a higher frequency, the antenna can simply be shortened to the correct length corresponding to a half wavelength at the new frequency. The new radiation pattern will have largely the same characteristics as the first because the current distribution is the same relative to a wavelength. The same principle holds true for loops, slots, and microstrip antennas as well.

A number of mechanisms can be used to change the effective length of resonant antennas, although some of these are more effective than others in maintaining the radiating characteristics of the original configuration. The following describe different reconfiguration mechanisms, provide some examples, and discuss the benefits and drawbacks of each approach.

- Switches

The effective length of the antenna, and hence its operating frequency, can be changed by adding or removing part of the antenna length through electronic, optical, mechanical, or other means. There are different kinds of switching technology, such as optical switches, PIN diodes, FETs, and radio frequency microelectromechanical system (RF-MEMS) switches, in frequency-tunable monopole and dipole antennas for various frequency bands.

- Variable Reactive Loading

The use of variable reactive loading has much in common with the switched based reconfigurability. The only real difference between the two is that, in this case, the change in effective antenna length is achieved with devices or mechanisms that can take on a continuous range of values (typically capacitance) that allows smooth rather than discrete changes in the antenna's operating frequency band.

- Structural/Mechanical Changes

Mechanical rather than electrical changes in antenna structure can deliver larger frequency shifts, whether used for switched or continuously variable bands. The main



challenges with these antennas lie in the physical design of the antenna, the actuation mechanism, and the maintenance of other characteristics in the face of significant structural changes.

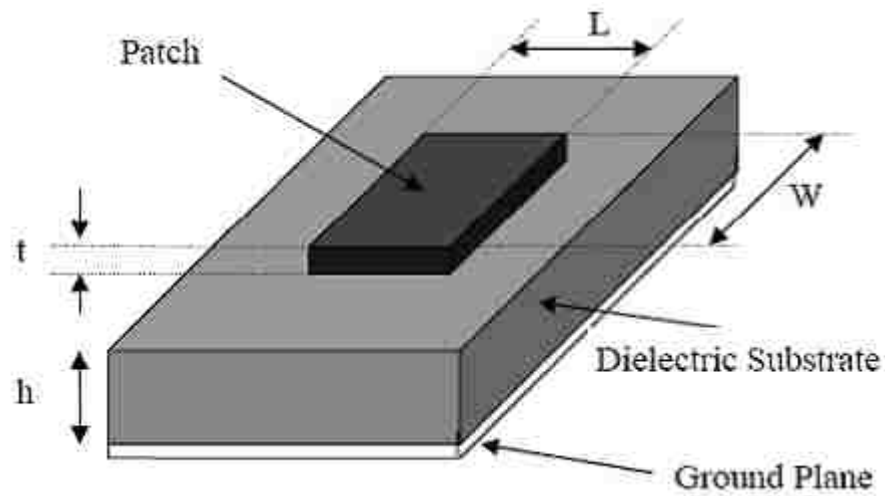
- Material Changes

Although changes to the conductors predominate in reconfigurable antenna designs, changes in the material characteristics of designs also promise the ability to tune antennas in frequency. In particular, an applied static electric field can be used to change the relative permittivity of a ferroelectric material, and an applied static magnetic field can be used to change the relative permeability of a ferrite. These changes in relative permittivity or permeability can then be used to change the effective electrical length of antennas, again resulting in shifts in operating frequency.

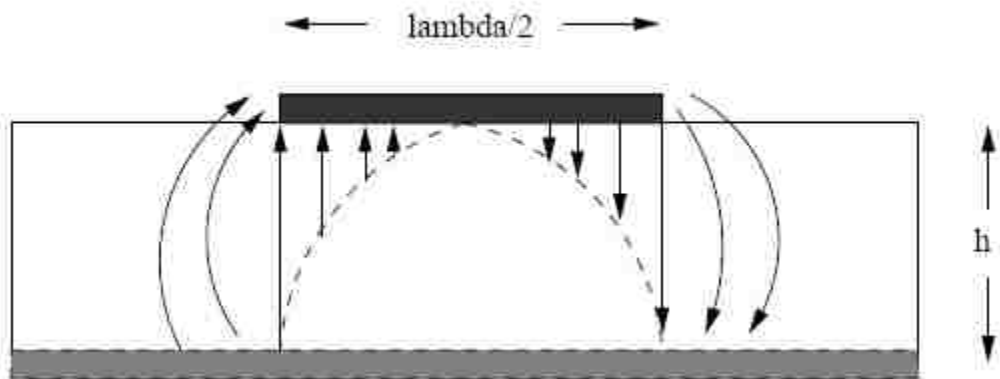
### 2.2.3 Microstrip Patch Antennas

Microstrip Patch Antennas (MSAs) became very popular in 1970s for space borne application [19]. Since then a massive amount of research and development efforts have been put into it. They are considered to be low profile, conformable to planar and non-planar surfaces, simple and inexpensive to fabric using modern printed-circuit technology, very versatile in terms of resonance frequency, polarization pattern and impedance. We can find MSAs on the surfaces of high-performance aircraft, spacecraft, satellites, missiles, cars and new held mobile telephone, where size, weight, cost, performance and ease of installation are important factors.

A conventional MSA consists basically of a radiating metallic patch on one side of a dielectric substrate, and has ground plane on the other side. Figure (2.6) shows top and side views of a conventional MSA.



(a) top view



(b) side view

Fig. 2.6: Structure of a conventional rectangular MSA: (a) top view, (b) side view.

The metallic patch can generally be made of conducting material such as copper or gold, and it can take many different configurations such as square, rectangular, dipole, circular, elliptical, triangular, disk sector, circular ring, and ring. However, the conventional square, rectangular and circular MSAs are the most popular because of simple fabrication, performance prediction and analysis, besides their attractive radiation characteristics such as low cross-polarization radiation.

Rectangular patches are probably the most utilized patch geometry. It has the largest impedance bandwidth compared to other types of geometries. Circular and elliptical shapes are slightly smaller than of rectangular patches. Thus it will have smaller bandwidth and gain. This circular geometry patches were difficult to analyze due to its inherent geometry.

Generally, MSA as shown in figure (2.7) consists of a very thin metallic strip (radiating patch)  $t$  placed a small fraction of a wavelength ( $t \ll \lambda_0$ , where  $\lambda_0$  is the free space wavelength) placed on one side of a dielectric substrate with thickness  $h$  ( $h \ll \lambda_0$ , usually  $0.003\lambda_0 \leq h \leq 0.05\lambda_0$ ) that locates above a grounded plane and below the strip. There are many kinds of substrates, which have different dielectric constants in the range of  $2.2 \leq \epsilon_r \leq 12$ , that we can use for designing MSAs [17]. For good antenna performance, thick dielectric substrates having dielectric constants in the lower end of the range mentioned above is desirable to get better radiation efficiency, larger bandwidth, offers lowest losses and loosely bound fields for radiation into space. This is besides enhancing the fringing fields which is account for radiation. However, such a

configuration leads to a larger antenna size. Besides, higher  $\epsilon_r$  results in smaller elements which boarder radiation pattern. However, very thick MSAs in spite of giving better bandwidth, it cases probe inductance that prevents patch impedance matching to the input connector, the probe inductance can be tuned out with a capacitive gap.

In MSA the radiating patch and the feed lines are usually photo etched on the dielectric substrate, it radiates primarily because of the fringing fields between the patch edge and the ground plane as shown in figure (2.8).

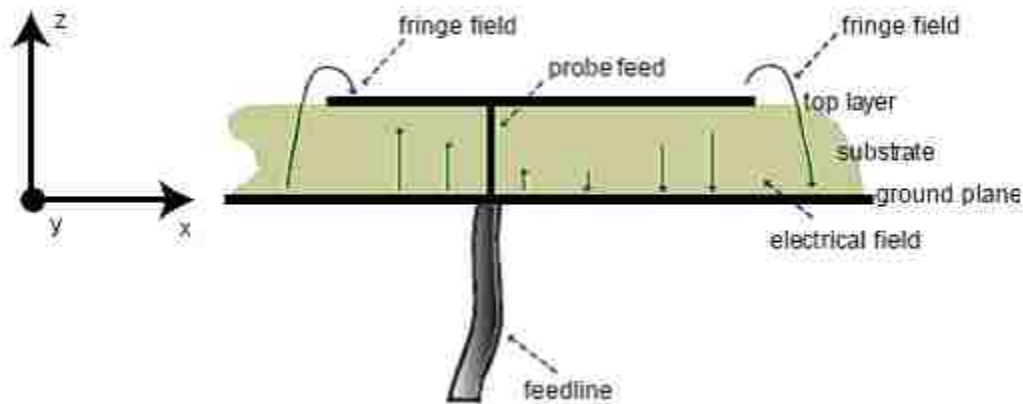


Fig. 2.7: Microstrip feeding and fringing field lines.

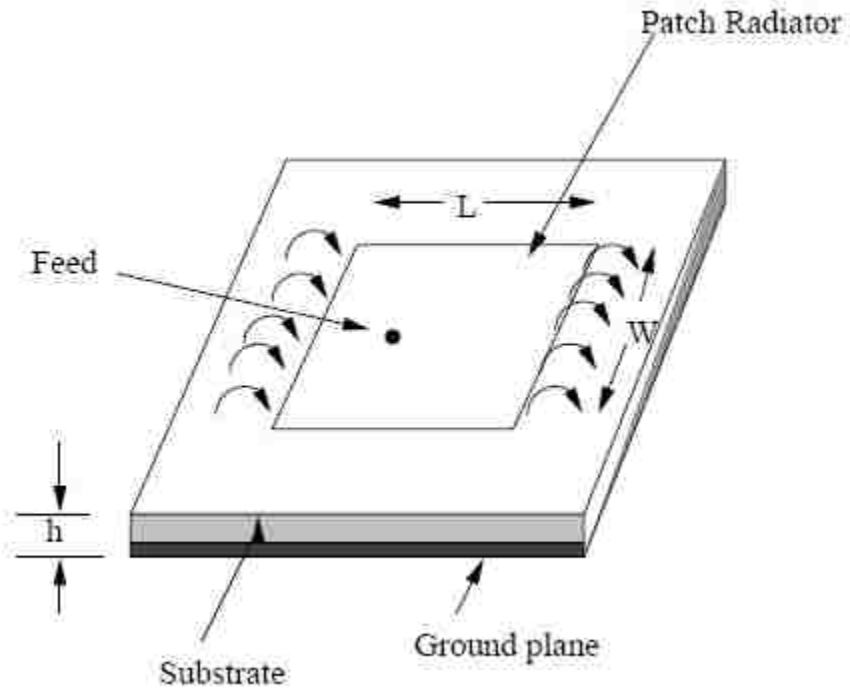


Fig. 2.8: Microstrip fringing field lines.

## 2.3 Field Programmable Gate Arrays

The configuration of the antenna is accomplished by utilizing a Field Programmable Gate Array (FPGA). FPGAs are constructed using one basic “logic-cell”, duplicated thousands of times [25]. A logic-cell is simply a small lookup table (LUT), a D flip flop, and a two to one multiplexer for bypassing the flip flop.

The LUT is just a small Random Access Memory (RAM) cell and usually four inputs, so it can in effect become any logic gate with up to four inputs. For example, an AND gate with three inputs, whose result is then sent through an OR gate with some other input would be able to fit into one LUT.

Every logic cell can be connected to other logic cells using interconnect resources. Interconnect resources are wires and multiplexers that are placed around the logic cells. While only one logic cell cannot accomplish much, lots of them connected together can accomplish complex logic functions. The FPGA's interconnect wires extend to the boundary of the device where Input Output cells are used to connect to the pins of the FPGA.

Besides the general purpose interconnect resources, FPGAs contain fast dedicated lines connecting neighboring logic cells. A technology somewhat unique to FPGAs, programmable technologies such as PAL and CPLD do not contain fast dedicated lines. These lines were implemented order to create arithmetic functions like counters and adders efficiently.

### 2.3.1 JTAG 1149.1 Standard

Voltages are asserted on four output lines from the FPGA by way of the Joint Test Access Group 1149.1 standard [26]. This standard started being developed in 1985 as part of the Joint European Test Action Group. The group collaborated with North America, became JTAG, and submitted a series of proposals for a standard form of boundary scan to the IEEE. The IEEE initially approved the standard in February 1990.

The standard can be used to assert signals, but the standard's main function is the boundary scan. Without using physical test probes, the boundary scan methodology allows one to test circuitry, interconnects, and cells of logic. By creating test cells which are then joined to every pin on the device, boundary scan can assert signals on specific pins by selection. The test cells are toggled using the JTAG serial scan chain line Test Data In (TDI). As outputs change on other pins, the test cell at that location can be read as Test Data Out (TDO). Thus, it is possible to verify proper circuit function. For example, if the circuit is shorted to another pin, the signal will not make it to the proper output pin and the short will be recognized.

However, while using this technique on integrated circuits, test cells must be inserted around logic blocks in order to isolate them as separate circuits from the rest of the device.

A major advantage of using JTAG is that it will not interfere with circuit function when not in testing mode. While in testing mode however, specific test conditions can be chosen. In fact, Xilinx implements a proprietary version of a JTAG Test Access Port controller on their FPGAs in order to program the device. A serial scan chain is sent to

the TAP controller and the logic cells and interconnects on the device are programmed using this information.

### 2.3.2 TAP Controller

A Test Access Port (TAP) controller module is programmed in VHSIC Hardware Description Language (VHDL) onto the FPGA. The TAP controller consists of five single-bit connections as can be seen in Table 2.1:

Table 2.1: Test Access Port Connections

Abbreviation	Name	IO	Description
TCK	Test Clock	Input	provides the clock for testing
TMS	Test Mode	Input	used to select testing modes
TDI	Test Data	Input	Input line for sending data into chip
TDO	Test Data Out	Output	line for reading data out of the chip
TRDT	Test Reset Signal	Input	Input used to reset the TAP controller

The boundary scan uses a couple methods to make sure that the TAP controller is secure [27]. In order to prevent the boundary scan from running and allow the chip to run as designed, TRST\* and TCK are held low and TMS is held high. Another facet of the controller is to sample the inputs on the rising edge of the TCK clock and ensure that the outputs are produced on the falling edge of the clock so that race conditions are avoided.

A general boundary scan testing architecture is shown in figure (2.9). Several items of hardware make it up:

- The five lines that are part of the TAP interface.
- For obtaining data from the device under test there are data registers (DRs)



- The instruction register (IR) dictates what the TAP controller should do
- The finite state machine or TAP controller, which is in charge of the inputting signal to the IR and DR

This finite state machine known as the TAP controller configures the system in two different modes. In one of its modes, the controller inserts an instruction into the instruction register to specify what boundary scan should do. In the other mode, it inserts data into and out of the data register. The IEEE standard demands the provision of at least two test data registers. These registers are known as the boundary scan register and the bypass register. The most important register for testing is the boundary scan register because it represents each input and each output on the device. The other register is known as the bypass register and is simply a single flip flop. It is also important since it is used to allow the TAP controller to skip the testing of idle devices and focus on testing one particular device.

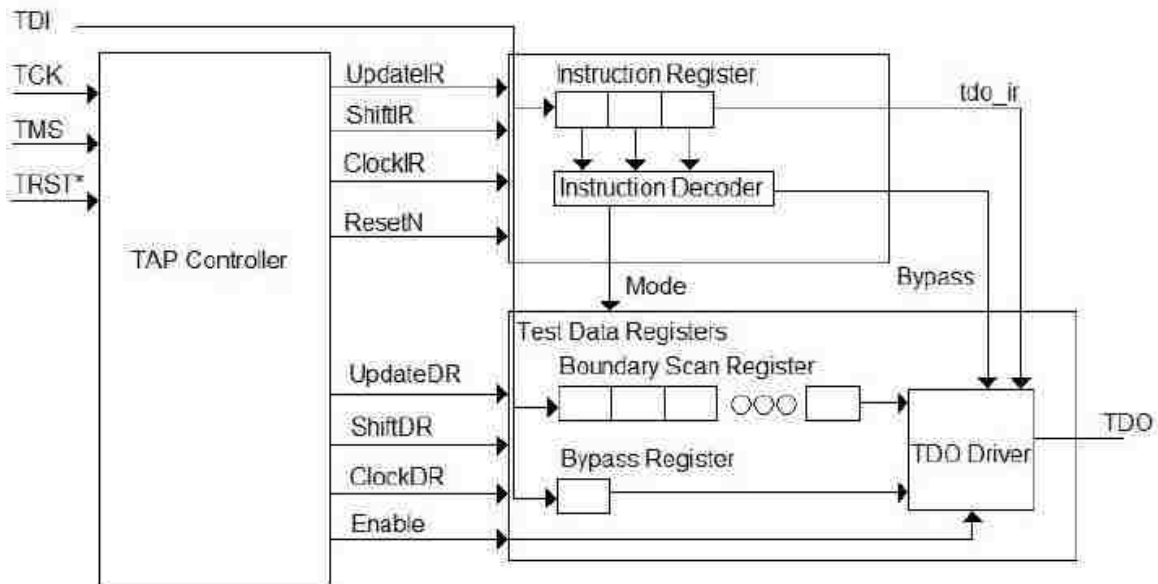


Fig. 2.9: Architecture of the TAP controller.

### 2.3.3 The Instruction Register

The instruction register (IR) must contain at least two bits because boundary scan employs two data registers: the bypass and boundary scan registers [27]. If the data register is chosen, the instruction register must select which of the data registers will be used for the boundary scan. In the CaptureDR state, the instruction register figures out where the data register will get its value from. It also determines if the values are to be sent to core logic or output cells. The designer of the TAP controller can specify instructions, but three instructions are necessary and two more are highly recommended:

The Bypass instruction puts the bypass register in the data register scan path in order to skip the testing of a specific device. This means that one flip flop passes along the bit stream from TDI to TDO. By using this instruction, a user can avoid devices that do not need testing and not have to send the bit stream scan through all the shift register stages of every device. It is recommended that this instruction be signified by ones in every cell of the instruction register.

The Sample/Preload instruction puts the boundary scan register in the data register scan path. This means that the bits the user wants tested on the inputs and outputs of the device are loaded into the data register. Sample/Preload simply copies the device IO values into the data registers in the CaptureDR state. Then the values can be subsequently moved out of the data register using the ShiftDR state. As this occurs, fresh values are loaded into the data registers but not activated on the inputs and outputs of the device.

The Ex-test instruction is a lot like Sample/Preload but makes it possible for the TAP controller to use boundary scan on the output cells of a device. This means that Ex-test can test the interconnecting lines between devices. Ex-test is accomplished by loading the values of the data register onto the output pads. Thus, by using a combination of ones and zeros on the output cells of one device and testing the input cells of another device, the connectivity between the devices can be determined.

The In-test instruction makes it possible to check circuitry within a device. In-test can more precisely pinpoint a discontinuity or short this way. By pacing through each part of a device, the core circuitry can be tested. The boundary scan registers are responsible for inserting ones and zeros not onto the input or output cells of the device, but into the core circuitry.

The RunBIST (Built In Self Test) instruction can trigger any Built In Self Tests that the device may be equipped with. While this instruction is dependent upon whether the device has a BIST or not, it can be invaluable in a thorough check of functionality.

The instruction register is made up of at least two implementations of IR bit. A shift register is created by linking two ClockIR flip flops. The bit stream inserts data into this shift register when the TAP controller is in the CaptureIR state. Then in the ShiftIR state, new data and thus another instruction is loaded. The instructions that get loaded are determined by the user. However, in order to check the integrity of the scan chain, the last two bits must be a 01 combination.

The UpdateIR state copies the instruction register data in parallel to the contents of the shift register so that the instruction register never contains an unrecognized

instruction. It is recommended that when the TAP controller is reset the instruction register get loaded with an instruction that will not mess up the function of the device or the TAP controller.

## CHAPTER III: NN-FPGA CONTROLLER DESIGN

### 3.1 NN Implementation

Implementing a neural network in reconfigurable architectures like FPGAs is an efficient way to calculate weights and network topologies. Network parameters are mapped into a hardware structure that improves the performance and the efficiency. The size of the implemented NN is limited by the block RAM capacity of the FPGA board.

The size of a NN is the amount of neurons synaptic weights available. Number of synaptic weight connections ( $N_{sw}$ ) available for FPGA board [12]:

$$N_{sw} = \frac{(\text{FPGA Block RAM Size})(\# \text{ of Block RAMs})}{\text{Size of Synaptic Weights}} \quad 3.1$$

To ensure effective utilization of FPGA board resources, a high efficient NN should be built. Network efficiency is the percentage of processing elements operating in parallel.

$$\text{Eff} = \frac{\# \text{ of neurons in the smallest layer}}{\# \text{ of neurons in the largest layer}} \times 100\% \quad 3.2$$

Maximum network efficiency is achieved when the network layers have the same number of neurons. Thus, to increase the efficiency, neurons in the largest layer are multiplexed in groups.

### 3.2 NN modeling on FPGA procedure

The controller design starts in building and training a neural network model for the antenna using Matlab and XSG, then, Xilinx ISE, the model is sent to an FPGA board.

The whole process (shown in figure (3.1)) can be summarizing in the following steps:

1. Measured antenna S11 data are collected, sampled, and normalized.
2. Matlab m-code is written to build and train the NN.
3. A Matlab Simulink model is built for the NN using XSG blocks.
4. VHDL code for the design is generated using Xilinx System Generator.
5. The VHDL code is sent to ISE, where it synthesized, implemented, and sent to the FPGA board.
6. FPGA board output is connected to the antenna input.

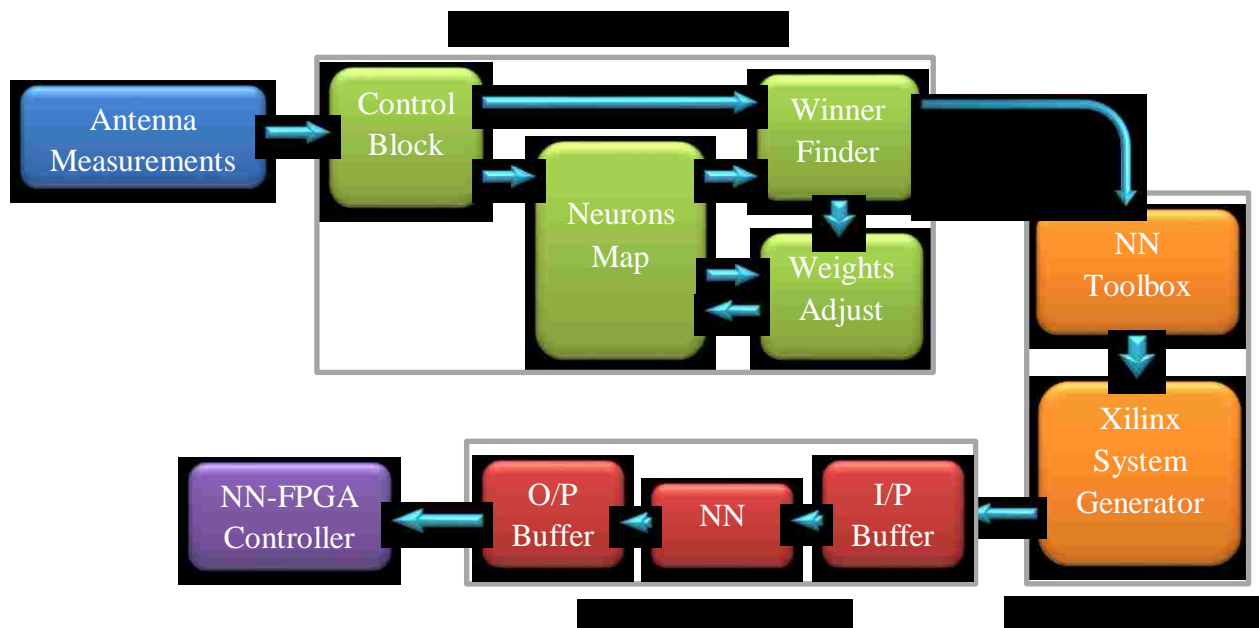


Fig. 3.1: NN modeling on FPGA procedure

Each of these steps will be explained thoroughly below.

1- Measured antenna S11 data are collected, sampled, and normalized:

The antenna output is measured using a Network Analyser (NA). Measured data are saved as a look up table for the NN; in this work, data were saved in an excel format. The output of the antenna (S11) is the input to the neural network, and the input of the antenna (switch configurations, or voltage level) represents the desired output from the neural network.

2- Matlab m-code is written to build the NN:

A Matlab code is written with help of NN toolbox. The code reads the xls data, then, builds an input/output arrays for the NN model.

Throughout this work, feedforward back-propagation neural network is used for modeling. In feedforward networks, the information moves in only one direction, forward, from the input nodes, through the hidden nodes and to the output nodes. There are no cycles or loops in the network. Here, feedforward networks are useful because they can handle high- dimensional inputs more easily.

The backpropagation neural network has the advantages of available effective training algorithms and better understood system behavior. It has a hierarchical design consisting of fully interconnected layers of propagating nodes, with one or more hidden layers

between input and output nodes. This network is designed to reveal the nonlinear relationships between complex input and output.

Then a NN model is formed with three layers:

- Input layer: with N neurons, where N is the number of points required to reproduce antenna's S11 with all resonance frequencies at all switches configurations.
- Hidden layer: a single hidden layer is used with a sigmoid activation function. Number of neurons in this layer is determined by try and error.
- Output layer: number of neurons in the output layer is equal to number of switches of the antenna.

3- A Matlab Simulink model is built for the NN using Xilinx System Generator blocks.

In order to generate a valid HDL code, XSG requires the Matlab model to be built in Matlab Simulink using Xilinx blocks. Therefore, the “.m” code is manually built block by block in Simulink. Fortunately, due to NN structure, most of the design is formed using adders and multipliers.

However, the Matlab Simulink library provided by XSG, has all the required blocks for the design of ANN except a few functions such as a sigmoid function.

Therefore, approximation for the sigmoid function is needed to implement the NN. Approximations for the sigmoid function were suggested before using a Taylor series [28], or polynomial functions [29]. However, these approximations dealt with the sigmoid



as a whole function. This resulted in either a complicated costly approximation to implement, or an approximation that has poor accuracy in parts of the function's curve.

Here, a new approach is presented. Instead of approximating the sigmoid as one function, it is approximated by deviding the sigmoid in multiple segments. Each segment is represented by a simple function.

As shown in equation (3.3), the sigmoid function is given by:

$$\varphi(x) = \frac{1}{1 + \exp(-ax)} \quad 3.3$$

The idea is to divide the domain  $x$  into segments. The more segments we use the more accurecy and a higher cost we will get. So the number of segments has to be optimized to an accepted level of accurecy with available FPGA board resouses. For this work, five segments was good for the antennas modeled and type of FPGA board used.

So, the sigmoid is divided into five segments as shown in figure (3.2) below.

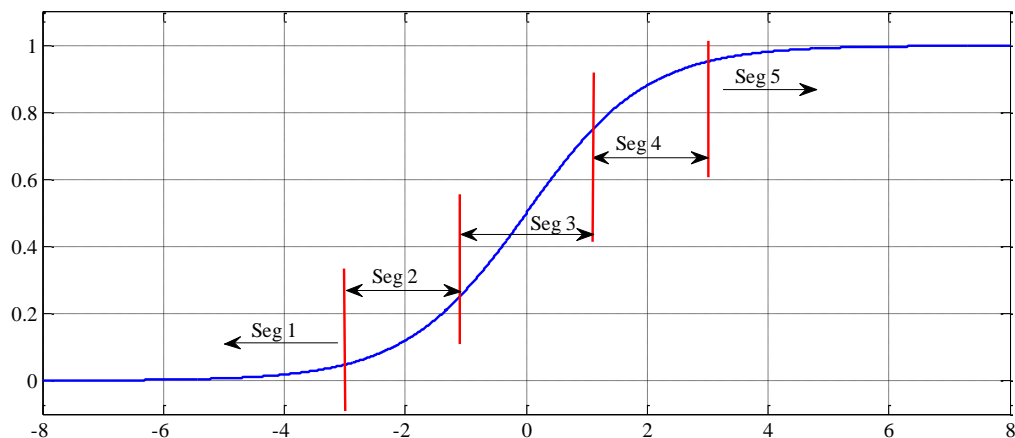


Fig. 3.2: Sigmoid function segments

The segments were as follows:

I. Segment 1:  $-\infty < x \leq -3$

In this interval, the function is approximated to have fixed value of 0.02

Segment's 1 function model is shown in figure (3.3).

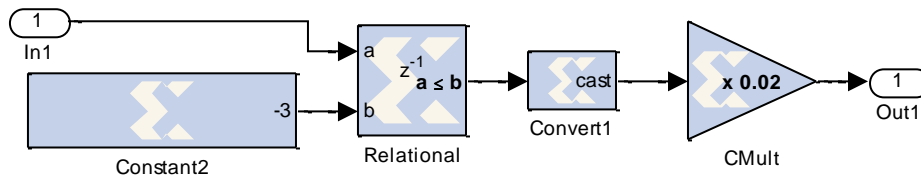


Fig. 3.3: Segment 1

II. Segment 2:  $-3 < x < -1.1$

In this interval, the function is approximated as a quadratic function:

$$\varphi_2(x) = 0.03809x^2 + 0.2617x + 0.4893 \quad (3.4)$$

Segment's 2 function model is shown in figure (3.4).

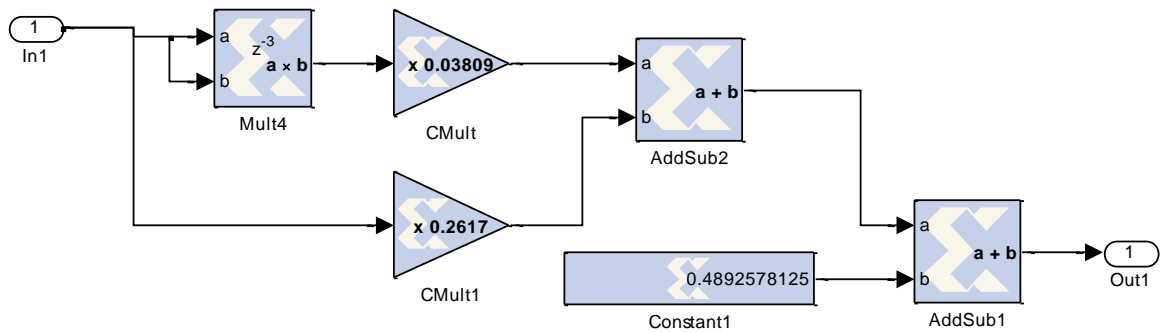


Fig. 3.4: Segment 2

III. Segment 3:  $-1.1 \leq x \leq 1.1$

In this interval, the function is approximated as a linear function:

$$\varphi_3(x) = 0.2314x + 0.5 \quad (3.5)$$

Segment's 3 function model is shown in figure (3.5).

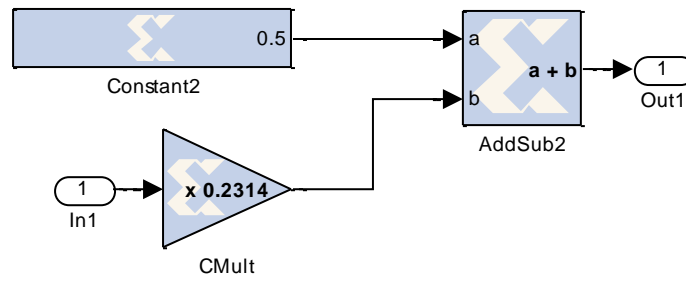


Fig. 3.5: Segment 3 function

#### IV. Segment 4: $1.1 < x < 3$

In this interval, the function is approximated as a quadratic function:

$$\varphi_4(x) = -0.03809x^2 + 0.2617x + 0.1507 \quad (3.6)$$

Segment's 4 function model is shown in figure (3.6).

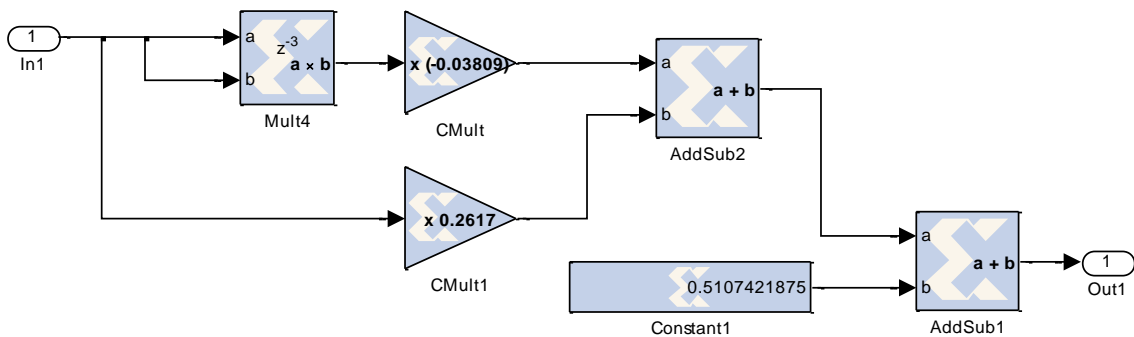


Fig. 3.6: Segment 4 function

V. Segment 5:  $3 \leq x < \infty$

In this interval, the function is approximated to have fixed value of 0.98 as shown in figure (3.7).

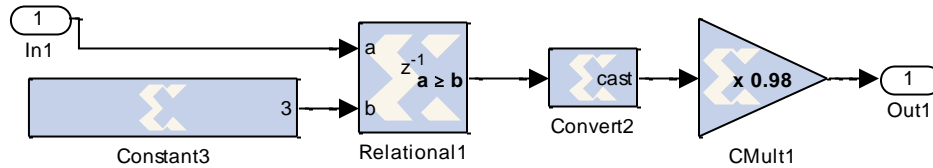


Fig. 3.7: Segment 5

The overall approximation function is:

$$\varphi_{\text{App}}(x) = \begin{cases} 0.02 & , -\infty < x \leq -3 \\ 0.03809x^2 + 0.2617x + 0.4893 & , -3 < x < -1.1 \\ 0.2314x + 0.5 & , -1.1 \leq x \leq 1.1 \\ -0.03809x^2 + 0.2617x + 0.1507 & , 1.1 < x < 3 \\ 0.98 & , 3 \leq x < \infty \end{cases} \quad 3.7$$

Figure 3.8 shows the whole sigmoid function Simulink model.

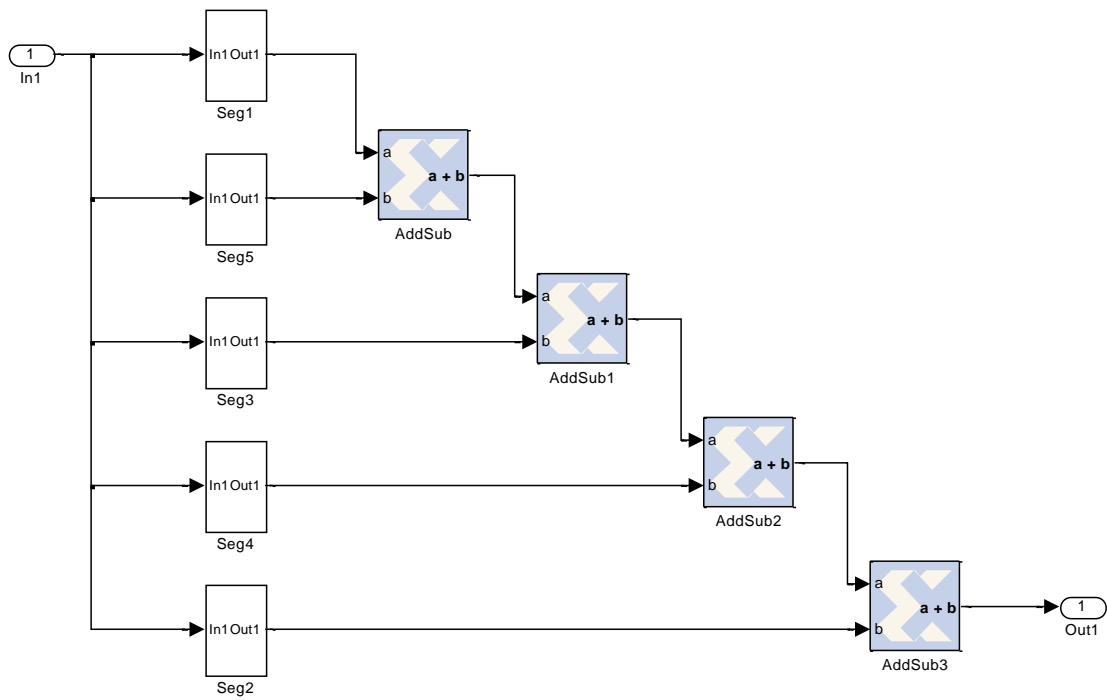


Fig. 3.8: Sigmoid Simulink model

4- VHDL code for the design is generated using Xilinx System Generator.

The System Generator block (Figure 3.9) provides control of system and simulation parameters, and is used to invoke the code generator. Code generator produces files consist of HDL, NGC, and EDIF that implements the design.

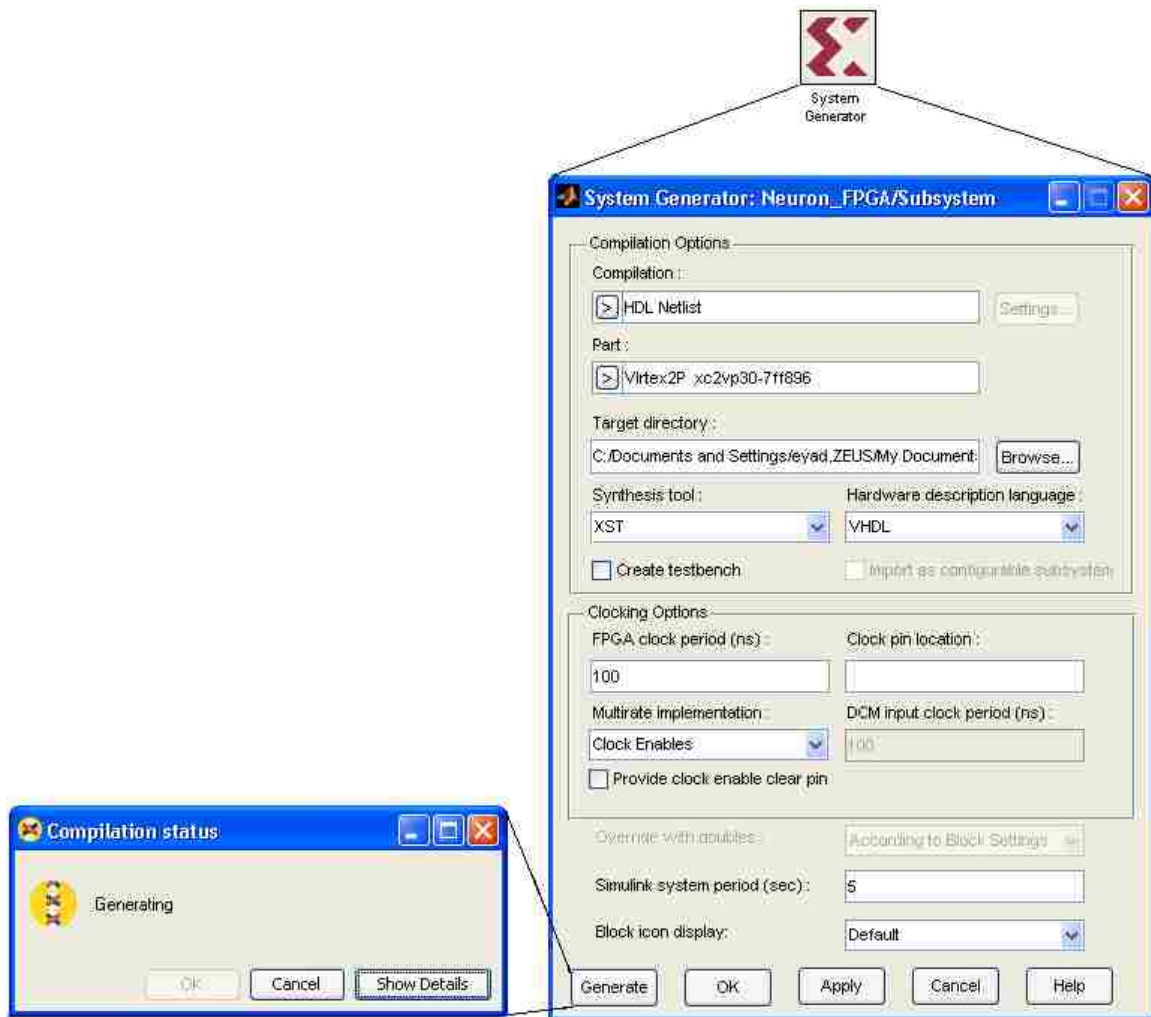


Fig. 3.9: System Generator block

- The VHDL code is sent to ISE:

These generated files can be opened using ISE project navigator, where it can be simulated, synthesized, and then transferred to the FPGA board.

- FPGA board output is connected to the antenna input. Figure (3.10) shows the actual FPGA board connections.



Fig. 3.10: Actual FPGA board connections

## CHAPTER IV: EXAMPLES, SIMULATIONS AND RESULTS

The designed NN-FPGA controller is applied to different reconfigurable antennas. For each antenna, a different NN-FPGA model is built and analyzed.

Simulations were done using Matlab R2009a and the Xilinx System generator 12.1 on an Intel core duo T5450 computer with 2GB RAM and 32-bit windows 7. FPGA board used was the Xilinx ML403.

### 4.1 Example 1: Reconfigurable Filtenna Antenna

A reconfigurable filtenna is an antenna that is tuned based on the integration of a reconfigurable filter in the antenna feeding line. It changes the operating frequency without incorporating active components in the antenna radiating surface. This can be implemented by integrating a reconfigurable band-pass filter within the feeding line of the antenna [30]. Thus the antenna is able to tune its frequency based on the filter's operation.

The filtenna structure consists of a dual sided Vivaldi antenna which is a wideband structure [31]. It is fed via a 50 ohms microstrip line which corresponds to a width of 5 mm. The antenna is made frequency reconfigurable by incorporating a band-pass filter in the antenna microstrip feeding line.



The antenna has a partial ground which is the ground plane of the filter of dimensions 30 mm x 30 mm. The structure is printed on a Taconic TLY substrate of dimension 59.8 mm x 30 mm. The inner and outer contours of the antenna radiating surface are based on an exponential function. The top layer constitutes the first side of the antenna radiating surface as well as the feeding line where the reconfigurable filter is located. On the bottom layer of the design resides the ground plane of the filter connected to the second radiating part of the Vivaldi antenna. The filtenna top and bottom layers are shown in Figure (4.1).

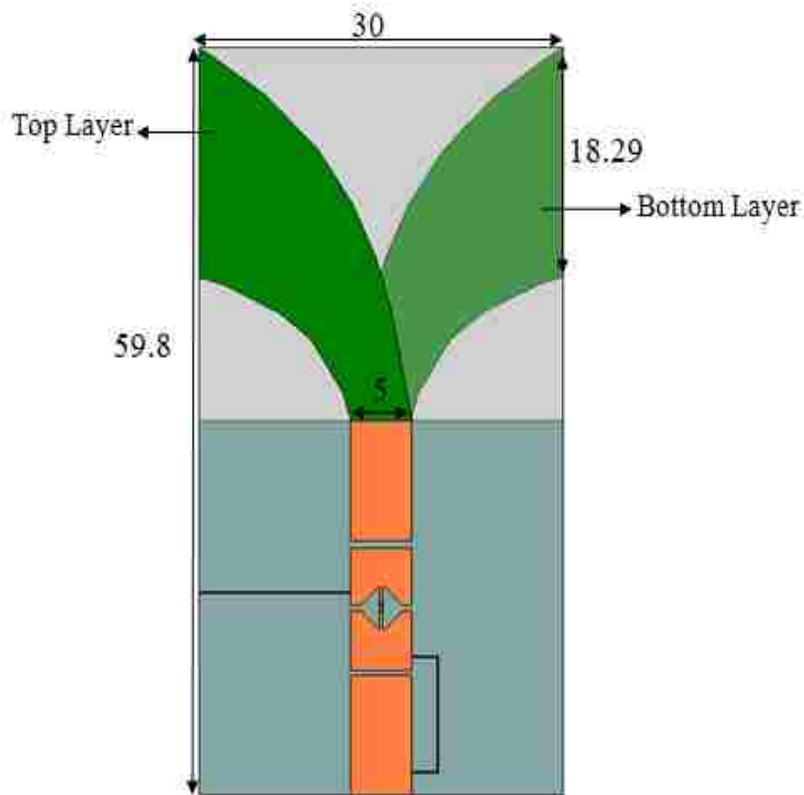


Fig. 4.1: Filtenna antenna structure

Table 4.1: Measured S11 for filtenna antenna

Frequency	Voltage	Min. S11
6.15E+09	11	-12.0981
6.20E+09	13	-12.7376
6.24E+09	15	-14.6652
6.29E+09	17	-14.5605
6.33E+09	19	-14.4031
6.37E+09	21	-15.5523
6.40E+09	23	-16.3904
6.43E+09	25	-17.4249
6.45E+09	27	-18.1084

For this antenna, S11 values were measured for input voltage range from 11 to 27, these data are shown in appendix B. For building NN model, only resonance frequency at each voltage level is needed, table (4.1) shows input voltages vs resonance frequencies measured.

As shown in figure (4.2), NN model of this antenna has one input neuron, one output neuron, and two hidden neurons. The input of NN is the desired resonant frequency of the antenna, and the output is the voltage level. NN parameters are shown in table (4.2) below.

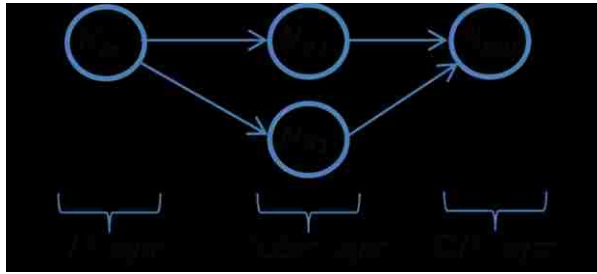


Fig. 4.2: Filtenna antenna NN model

Table 4.2: Neural network parameters for Filtenna antenna

Iterations	3
Input Neurons	1
Output Neurons	1
Hidden Layers	1
Hidden Neurons	2

Figure (4.3) shows the Simulink NN\_FPGA model of the antenna, layer 1 represents the hidden layer, while layer 2 is the output layer. Subsystems 1 and 2 are pre- and post- processing for input and output signals; it's mainly the normalization process.

Figures 4.4-6 show break down of the hidden and output layers.

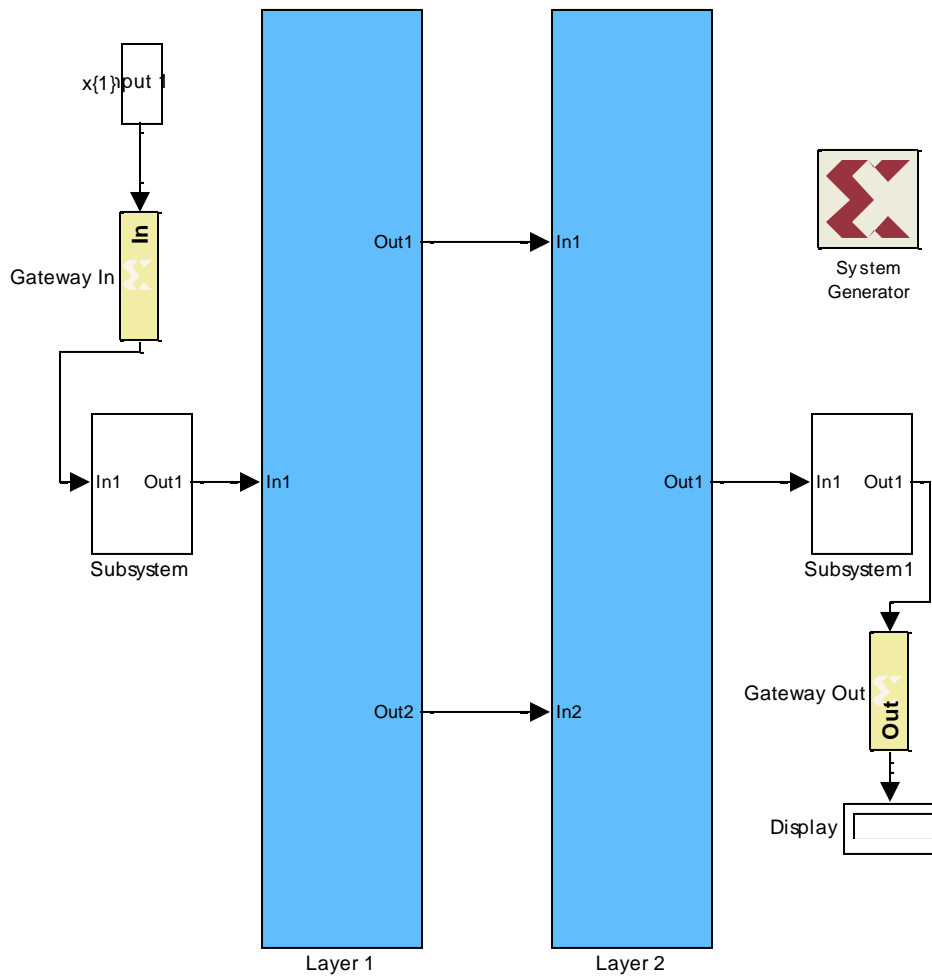


Fig. 4.3: Simulink NN\_FPGA model of the antenna

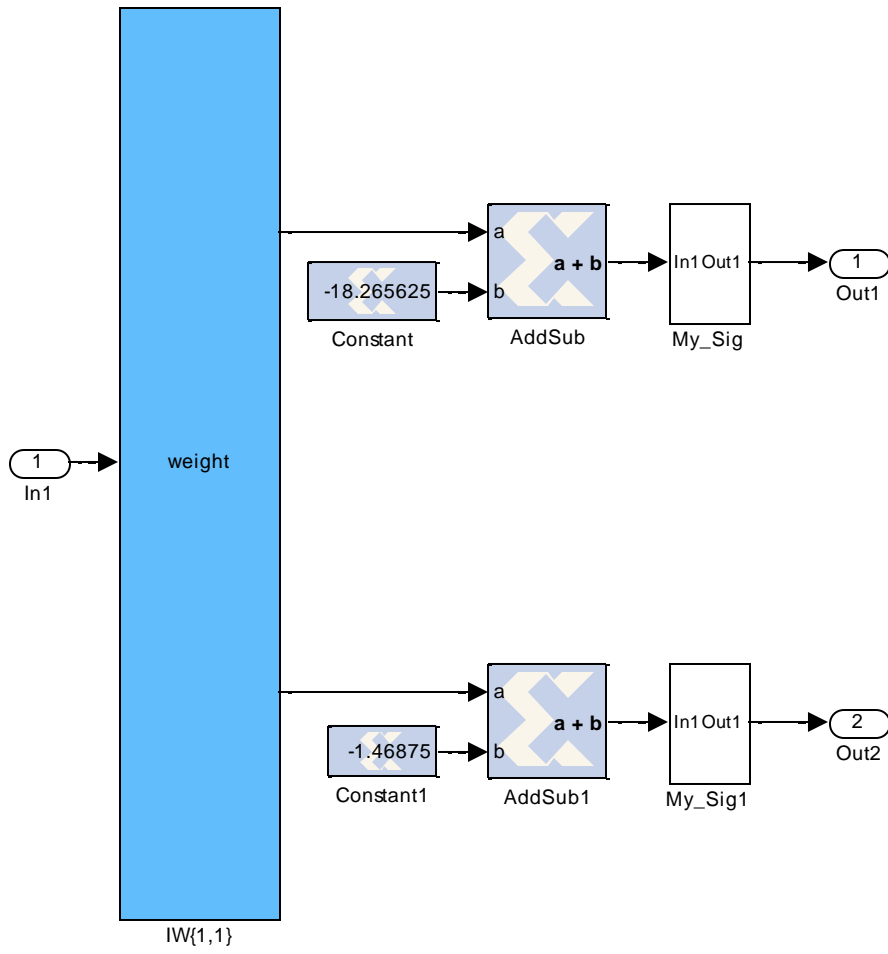


Fig. 4.4: Hidden layer of filtenna antenna

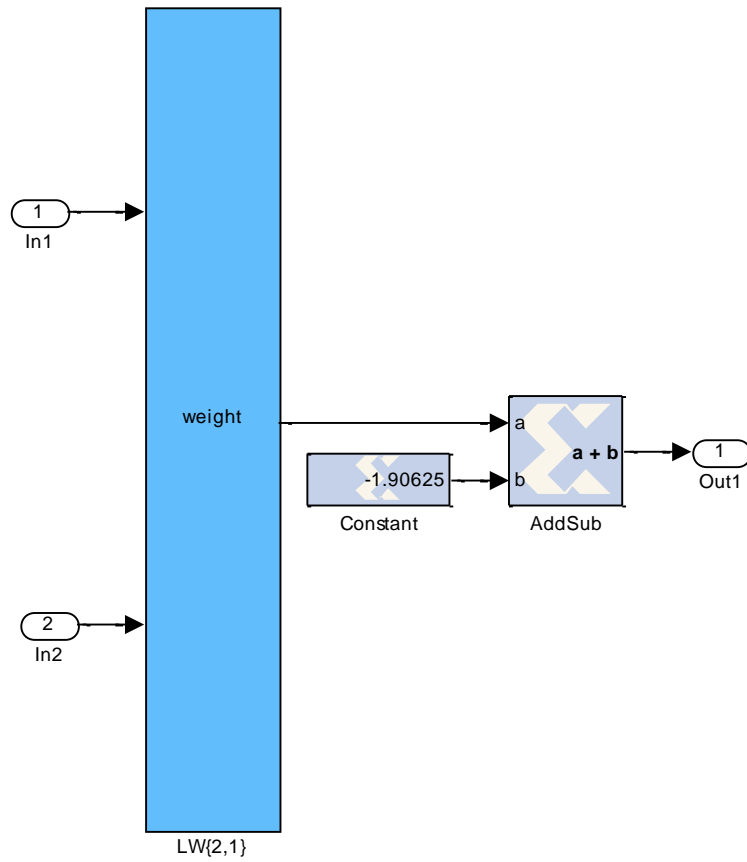


Fig. 4.5: Output layer of filter antenna

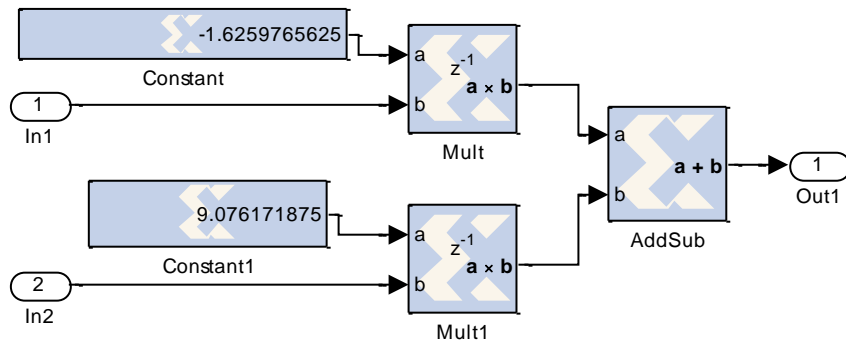


Fig. 4.6: output neuron structure

Figure (4.7) show the neural network output compared to the measured antenna response. Each point on the curve represents the resonance frequency at that specific input voltage. The training error was  $10^{-5}$ , however, due to approximations resulted from using Xilinx blocks and sigmoid function approximation, maximum error at FPGA level increased.

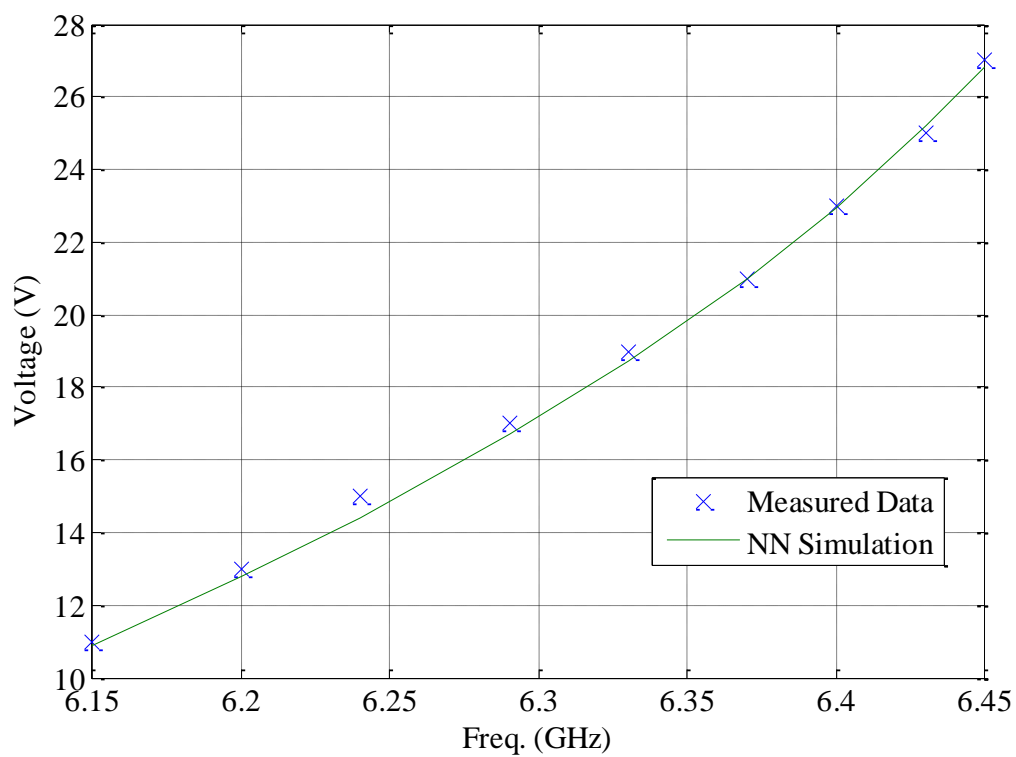


Fig. 4.7: NN output vs measured antenna response

Table 4.3: Filtenna NN simulation

Freq. (GHz)	Measured voltage (V)	NN simulated voltage (V)
6.15	11	10.89
6.2	13	12.78
6.24	15	14.41
6.29	17	16.69
6.33	19	18.74
6.37	21	20.97
6.4	23	22.97
6.43	25	25.23
6.45	27	26.82



## 4.2 Example 2: Microstrip Patch Antenna

As an example, a Neural Network modeling of a reconfigurable microstrip antenna [32] is presented. This antenna has 2 side slots. 1 switch bridge over each slot to achieve return loss tuning when activated and deactivated. By adjusting the status of the switches, the resonance frequencies can be varied, thus achieving frequency reconfigurability.

The resonance frequency ( $f_0$ ) of the patch is chosen to be around 2.4 GHz, and the dielectric material used is the RT/Duroid 5880 with a dielectric constant  $\epsilon_r = 2.2$ . The height of the ground plane is chosen to be 1mm, the thickness of the patch (t) is 0.7 mm and the height of the substrate (h) is 1.588 mm. The patch length and width and the ground plane length and width have the following values: L = 40 mm, W = 50 mm, Lg = 50 mm, Wg = 60 mm. This arrangement yields 3 resonance frequencies.

To get a reconfigurable antenna, one switch is added to each slot. Switch 1 (SW1) is fixed on the left slot at 20 mm above the lower edge of the patch shown in Fig. 1, and switch 2 (SW2) is fixed on the right slot at 16 mm below the upper edge of the patch. Both SW1 and SW2 are copper tape of 2 mm length and 1 mm width. The antenna structure, dimensions and switch placements are shown in figure (4.8).

Table 4.4: Neural network parameters for microstrip antenna

Iterations	6
Input Neurons	31
Output Neurons	2

Hidden Layers	1
Hidden Neurons	8

Antenna measurements are collected, sampled, normalized, and saved in xls data file. Training data for this antenna are shown in appendix B. Matlab m code reads the data, and places antenna switches possibilities as NN output, and antenna S11 as NN desired input. The network is designed as a back-propagation neural network. NN is designed with sigmoid activation function in the hidden layer, and linear activation function in the output layer.

As shown in table (4.4), neural network training takes 6 iterations to achieve the required accuracy with 31 input neurons (represent samples of S11), 8 hidden Neurons and 2 output neurons. Neural network representation for this antenna is shown in figure (4.9). NN is trained and weights are calculated. NN training error convergence is shown in figure (4.10).

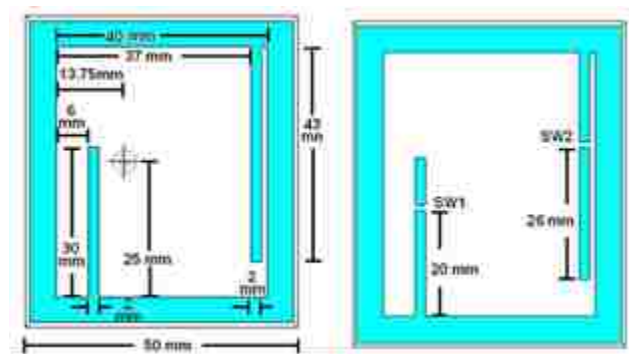


Fig. 4.8: Antenna Structure and dimensions where SW1 and SW2 represent the locations of each switch.

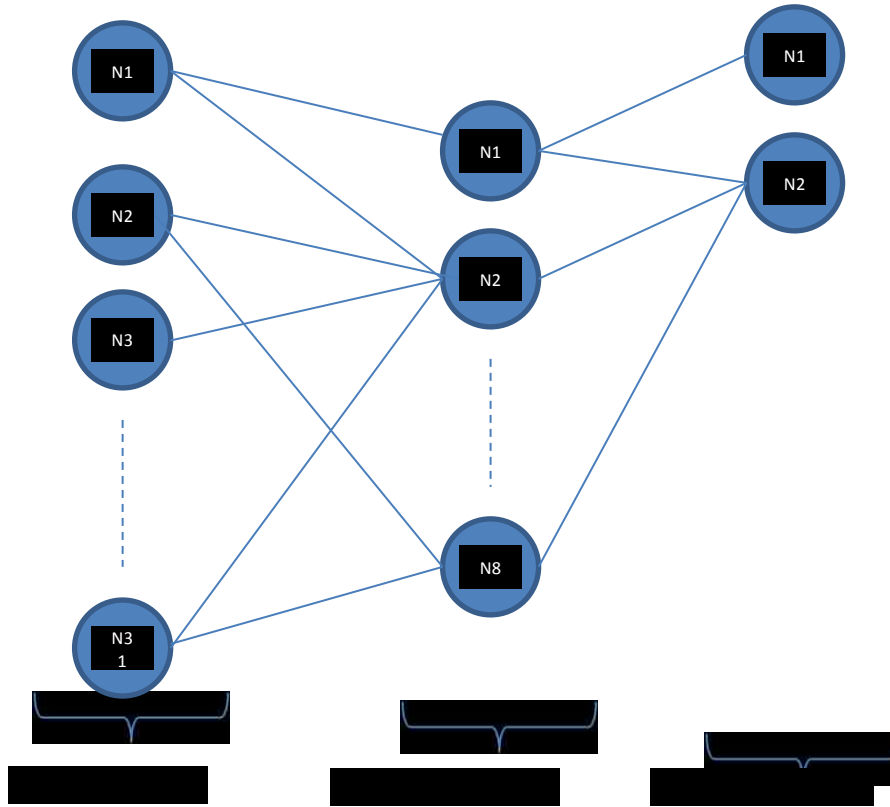


Fig. 4.9: Neural network representation for this antenna

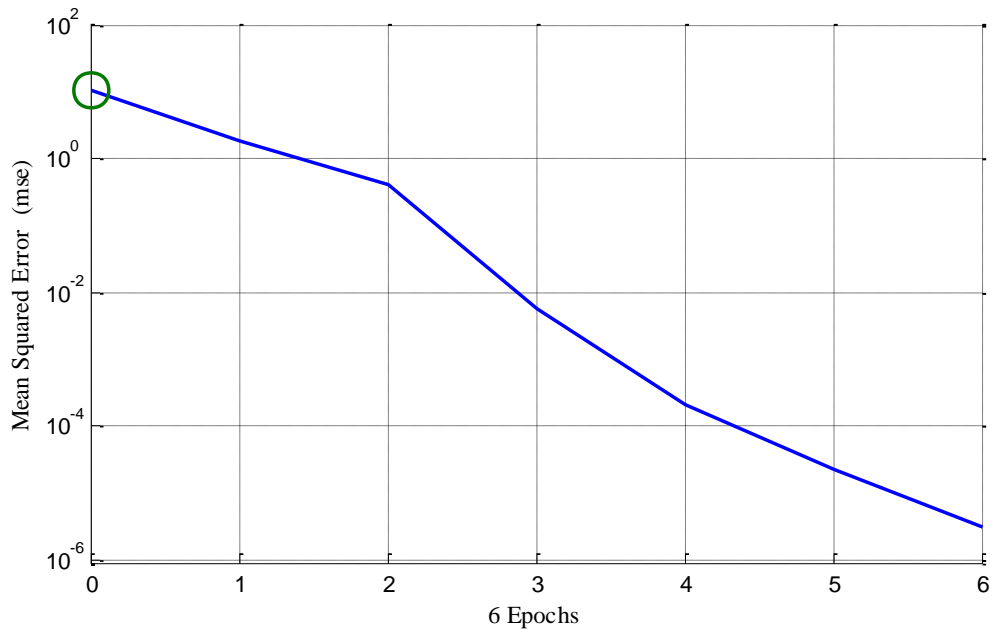


Fig. 4.10: NN training error

Number of hidden neurons is determined by training the NN in Matlab Simulink. For each run, the number of hidden neurons is changed and the output of the NN is compared to measured antenna response. Figure (4.11) shows NN output compared to measured antenna response for hidden neurons range from 2-10. The best matching was achieved at 8 hidden neurons.

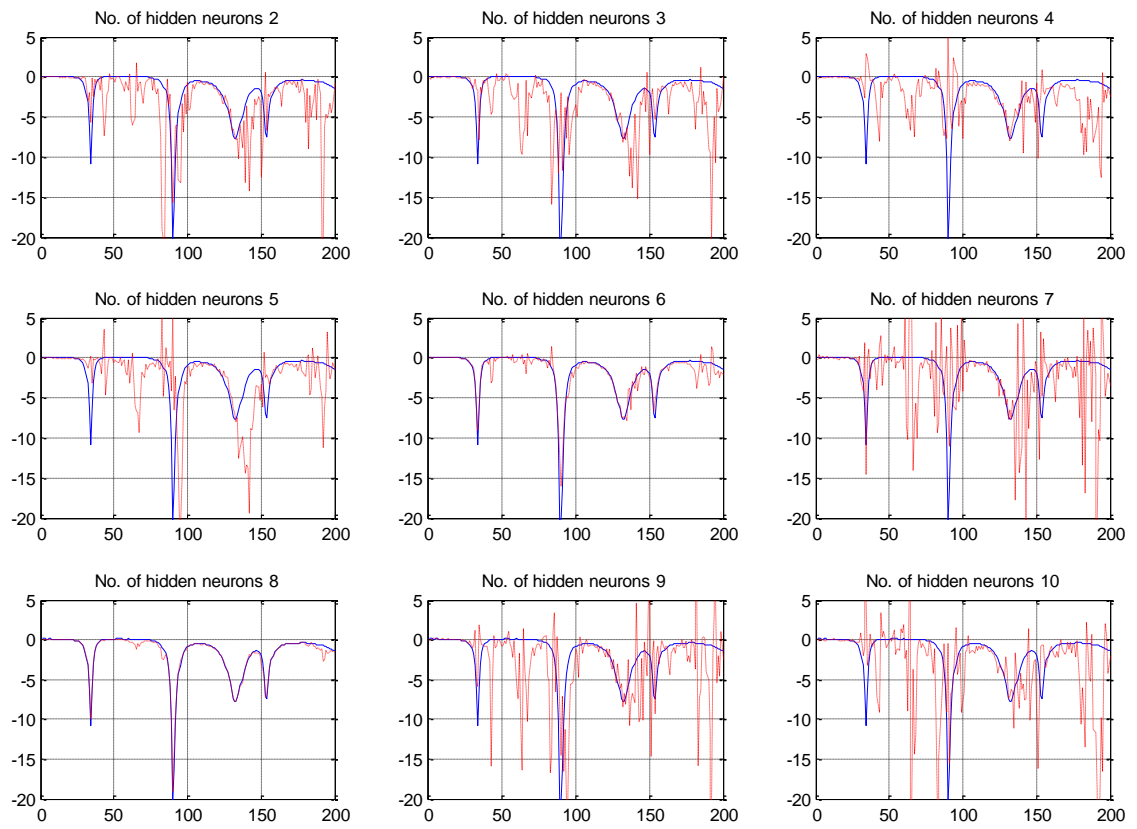


Fig. 4.11: Determining number of hidden neurons, NN output (red) vs. measured antenna response (blue)

Figures 4.12-15 show the neural network output compared to the measured antenna response for different switches configurations.

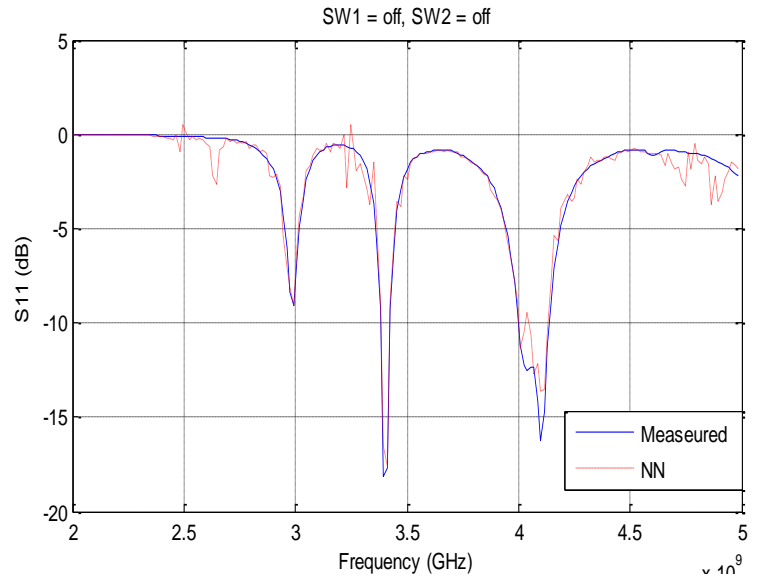


Fig. 4.12: NN output vs measured antenna response for SW1 off and SW2 off

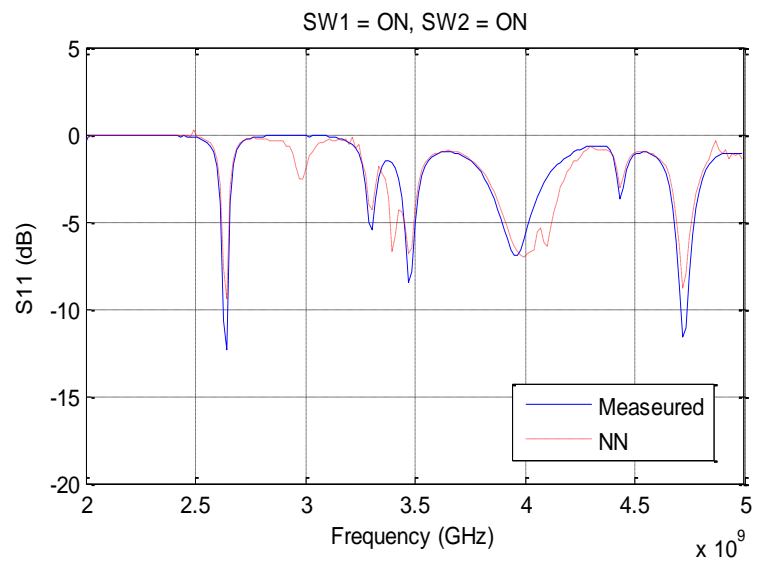


Fig. 4.13: NN output vs measured antenna response for SW1 on and SW2 on

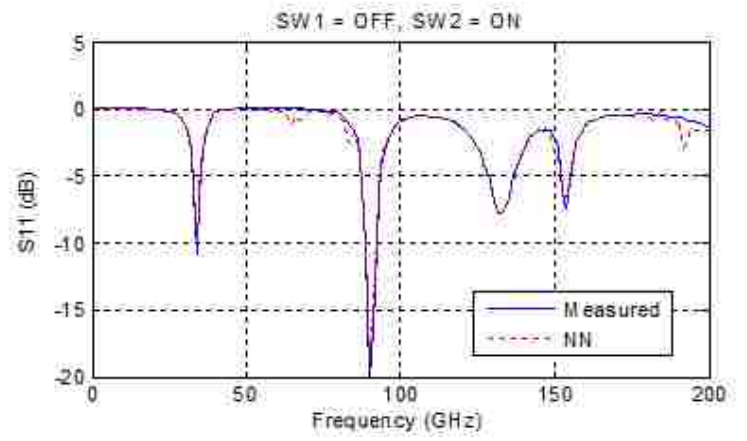


Fig. 4.14: NN output vs measured antenna response for SW1 off and SW2 on

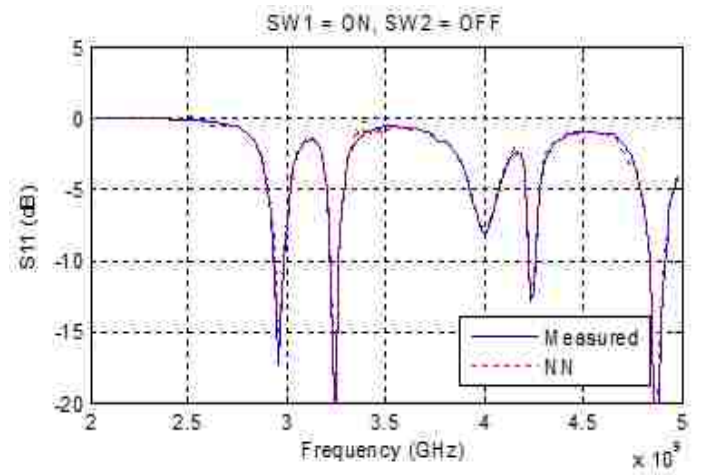


Fig. 4.15: NN output vs measured antenna response for SW1 on and SW2 off

### 4.3 Example 3: Rotatable Microstrip Antenna

In this example, a rotatable microstrip antenna [33] is modeled. Reconfigurability is achieved via a rotational motion of a part of the antenna patch while maintaining the same omni-directional radiation pattern in both the E and H planes. The rotating part has the form of a circle and contains four different shapes. Each shape corresponds to a different antenna structure. With every rotation, a different antenna structure is fed in order to produce a different set of resonant frequencies. Four different rotations can be done making the antenna cover five different bands (from 2 GHz up to 7 GHz) correspondingly.

The corresponding antenna structure is shown in Figure 4.16. It consists of two layers. The bottom layer is a partial ground to allow radiation above and below the substrate. The top layer is a rotating circular shape. The chosen substrate is Taconic TLY with a dielectric constant of 2.2 and a thickness of 1.6 mm. Figure 4.16 shows NN representation of the antenna.



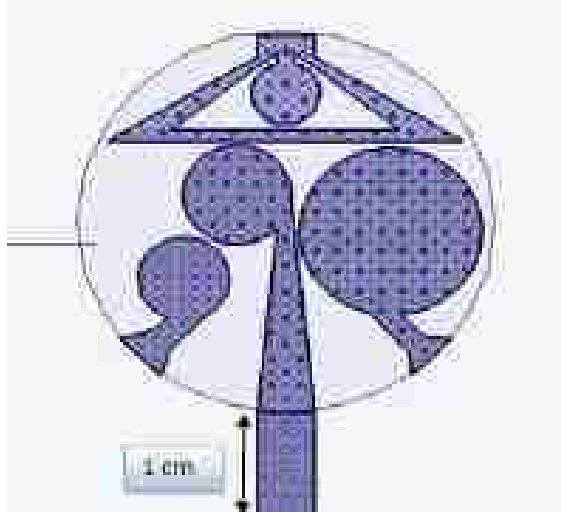


Fig. 4.16: Antenna Structure

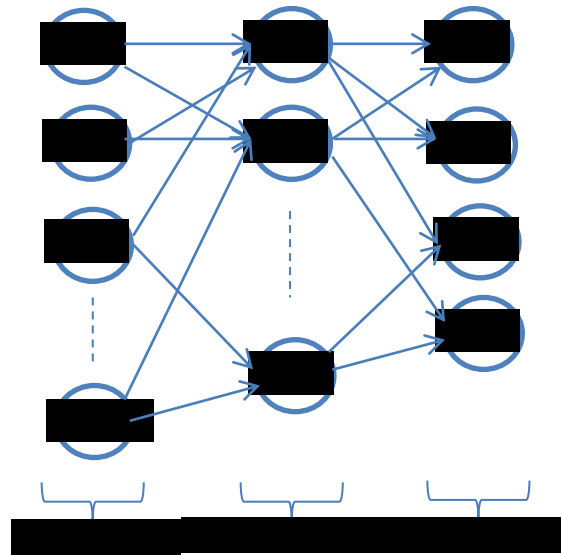


Fig. 4.17: Neural network representation for this antenna

Table 4.5: NN parameters of Rotatable Microstrip Antenna

Iterations	6
Input Neurons	201

Output Neurons	4
Hidden Layers	1
Hidden Neurons	7

Training data for this antenna are shown in appendix B. As shown in table (4.5), neural network training takes 6 iterations to achieve the required accuracy with 201 input neurons, 7 hidden neurons and 4 output neurons.

Figures 4.18-20 show the neural network output compared to the measured antenna response for different shapes.

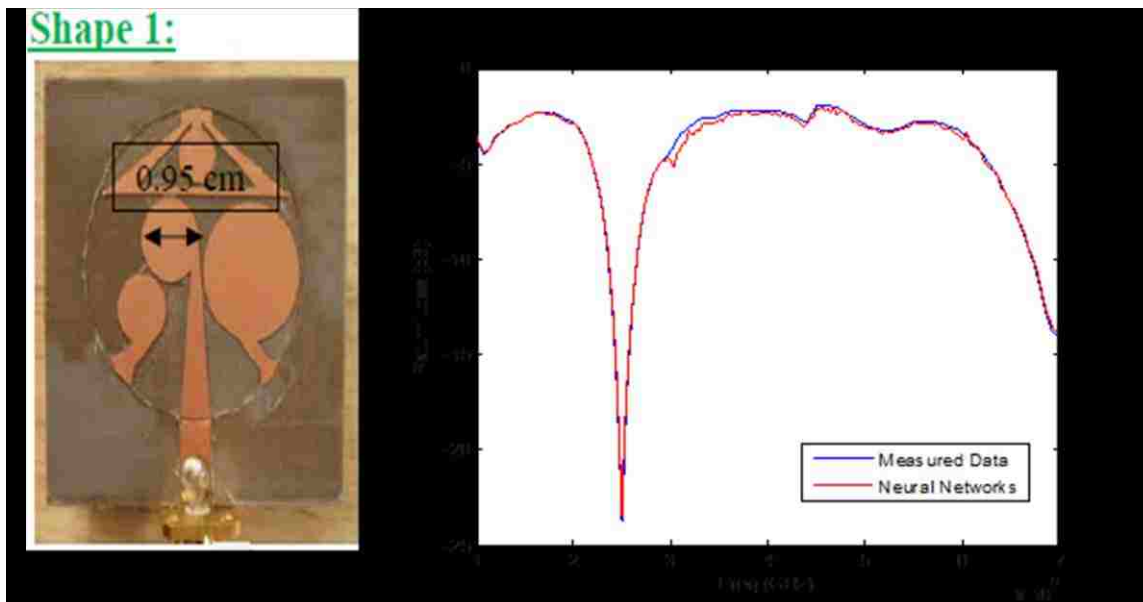


Fig. 4.18: NN output vs measured antenna response for shape 1

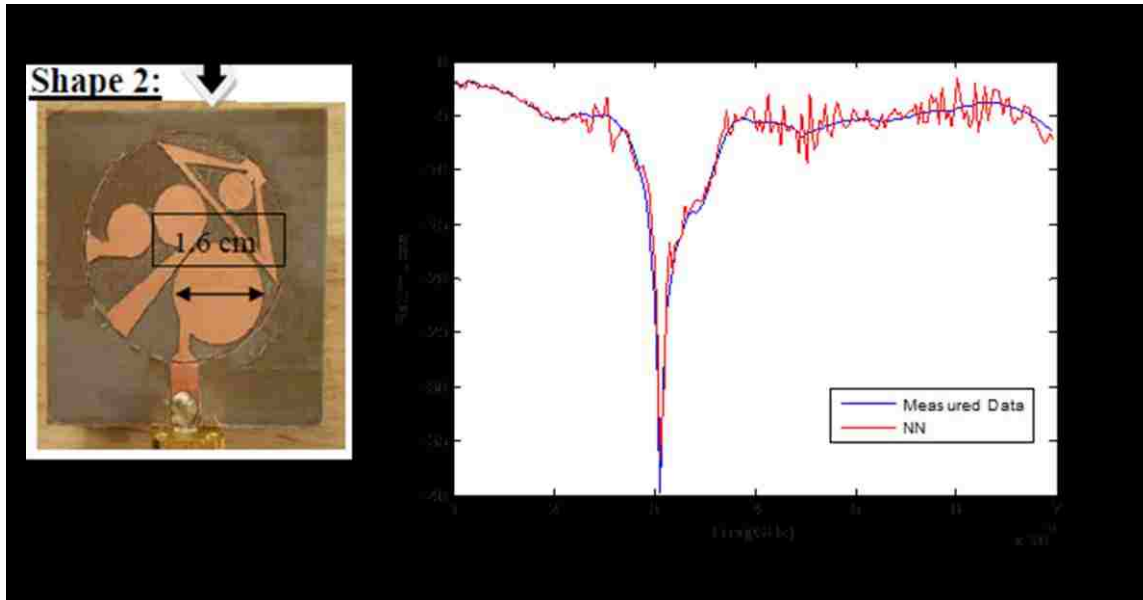


Fig. 4.19: NN output vs measured antenna response for shape 2

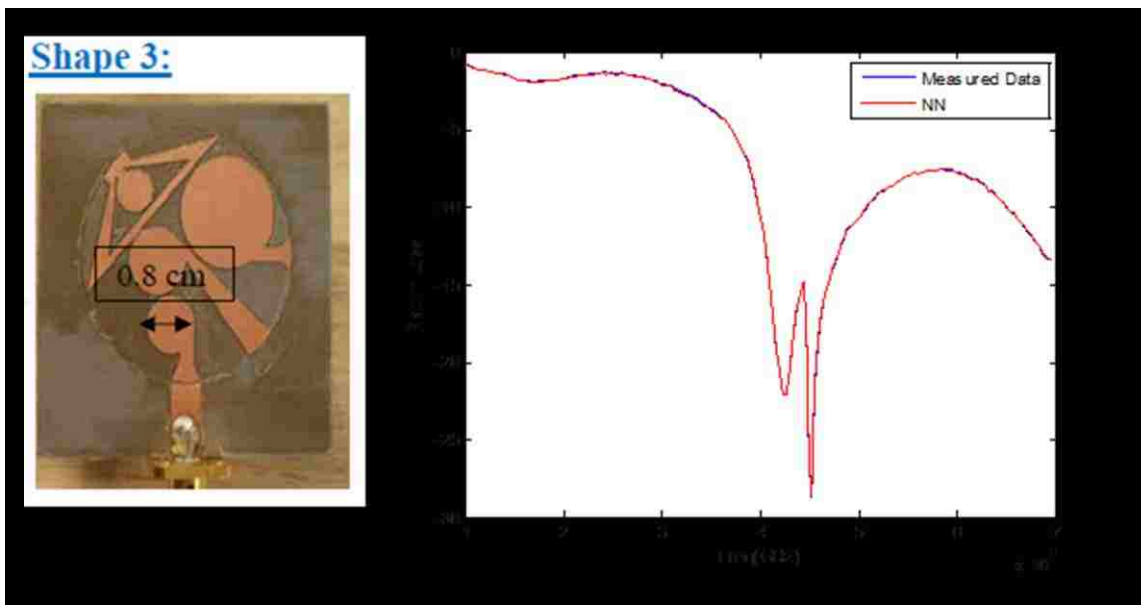


Fig. 4.20: NN output vs measured antenna response for shape 3

#### 4.4 Example 4: Planar Ultra-Wideband Antenna

The antenna is a monopole printed on a 1.6-mm-thick Rogers RT/duroid 5880 substrate with  $\epsilon_r = 2.2$  and features a partial ground plane [34]. Two circular split ring slots are etched on the patch, and two identical rectangular split rings are placed close to the microstrip line feed.

Figure 4.21 shows antenna structure and dimensions. Four electronic switches, S1, S2, S3 and S3', are mounted across the split ring slots and split rings. Switches S3 and S3' operate in parallel. That is, they are both ON or both OFF. As a result, eight switching cases are possible.

Whenever S1 is OFF, the larger split-ring slot behaves similar to a complementary split-ring resonator (CSRR) in causing a band notch. The size and location of this split-ring slot was optimized to have the notch in the 2.4 GHz band. Setting S1 to ON makes this notch disappear. Similarly, when S2 is OFF, a band stop is induced in the 3.5 GHz band, which disappears when S2 is ON. The two split rings near the feed line are designed so that a band notch is introduced inside the 5.2-5.8 GHz range when S3 and S3' are ON. The band stop is gone when S3 and S3' are OFF. The different cases lead to combinations of these notches. Case 8 features no notched bands, and thus corresponds to an ultra-wideband response covering the 2-11 GHz range, which is required for sensing.

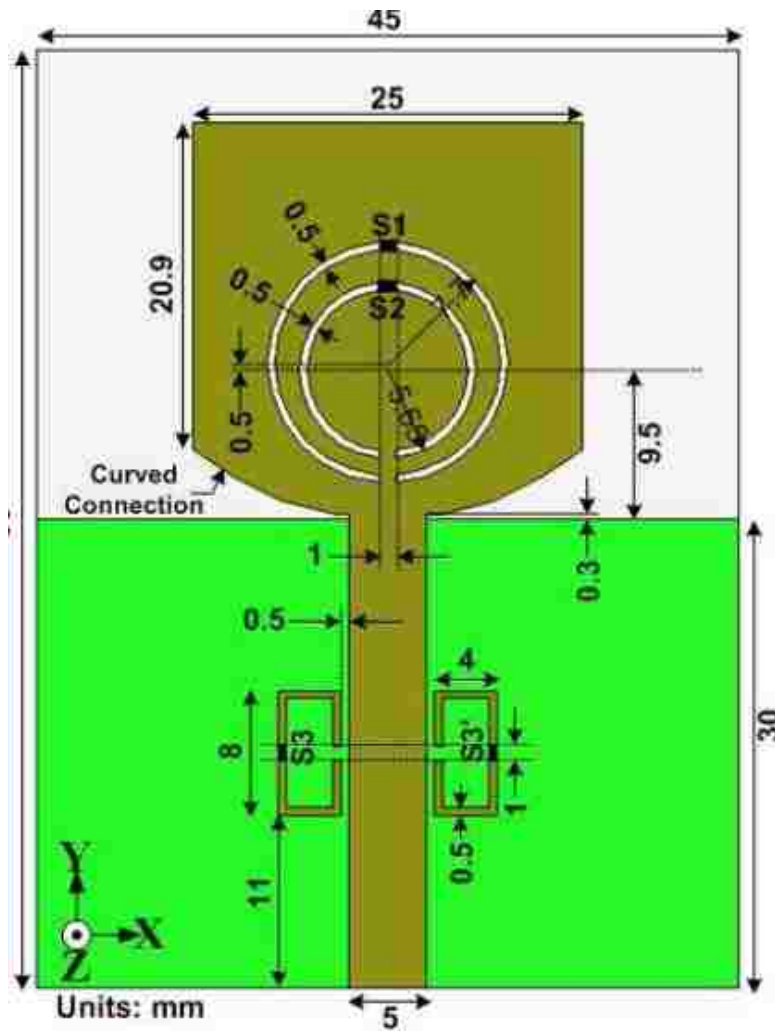


Fig. 4.21: Antenna Structure and dimensions

Table 4.6: NN parameters Planar Ultra-Wideband Antenna

Iterations	9
Input Neurons	201
Output Neurons	4
Hidden Layers	1
Hidden Neurons	8

Training data for this antenna are shown in appendix B. Table (4.6) shows that neural network training takes 9 iterations to achieve the required accuracy with 201 input neurons, 8 hidden Neurons and 4 output neurons.

Figures 4.22-25 show the neural network output compared to the measured antenna response for different cases.

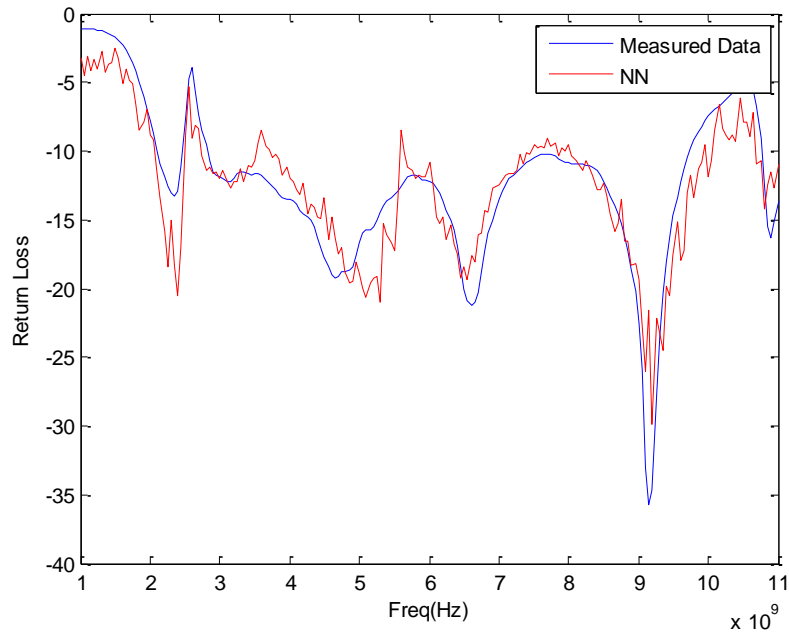


Fig. 4.22: NN output vs measured antenna response for case 0100

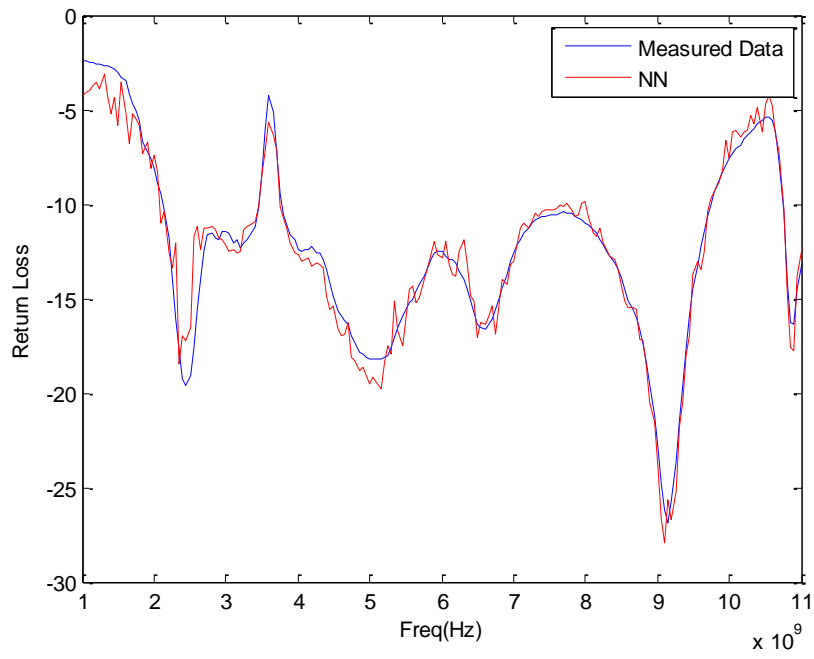


Fig. 4.23: NN output vs measured antenna response for case 1000

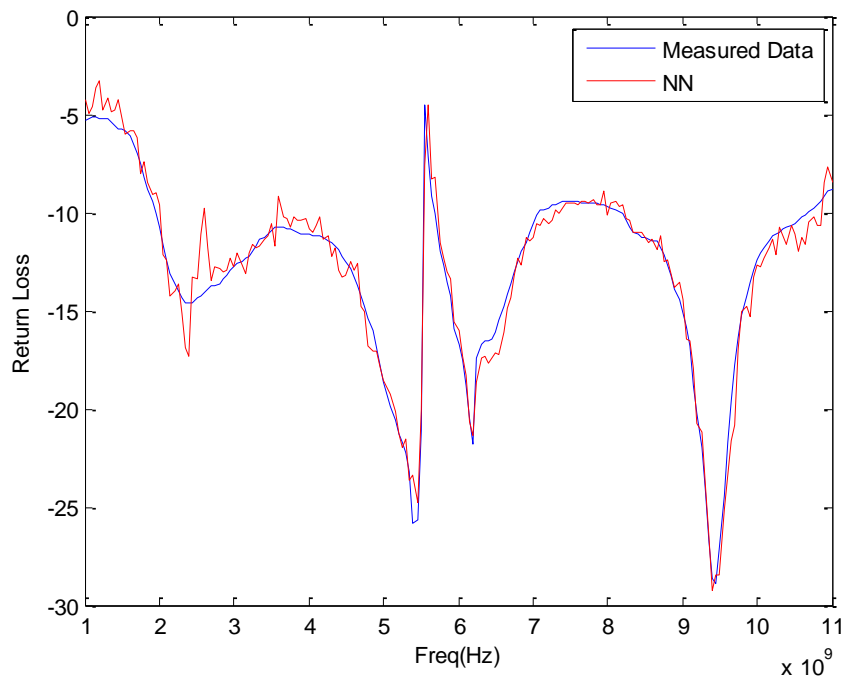


Fig. 4.24: NN output vs measured antenna response for case 1111

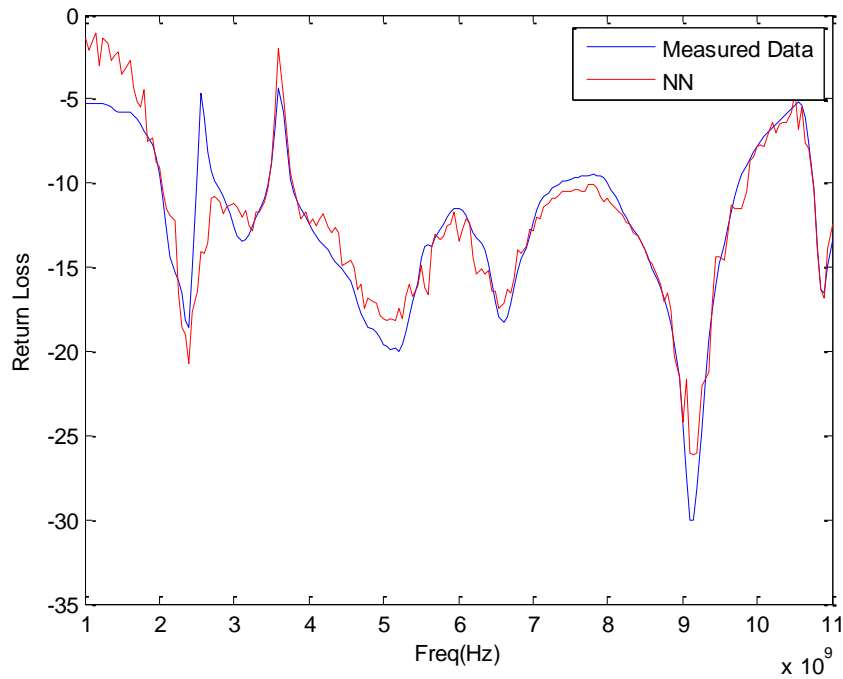


Fig. 4.25: NN output vs measured antenna response for case 0000

#### 4.5 Example 5: Reconfigurable filter

In this example, a reconfigurable defected microstrip structure (DMS) band pass filter is integrated with a broadband antenna [35]. The filter has a T-shape slot of dimension 2.25 mm x 2.8 mm and a coupling gap of dimension 0.2 mm. The purpose of the gap is to allow the filter to have the “band-pass” feature by functioning as a parallel-series resonance. The proposed design is printed on the Taclam plus substrate with a dielectric constant  $\epsilon_r = 2.1$  and a thickness 1.6 mm. The fabricated prototype and the dimensions of the T-shape DMS band pass filter are shown in figure (4.26).



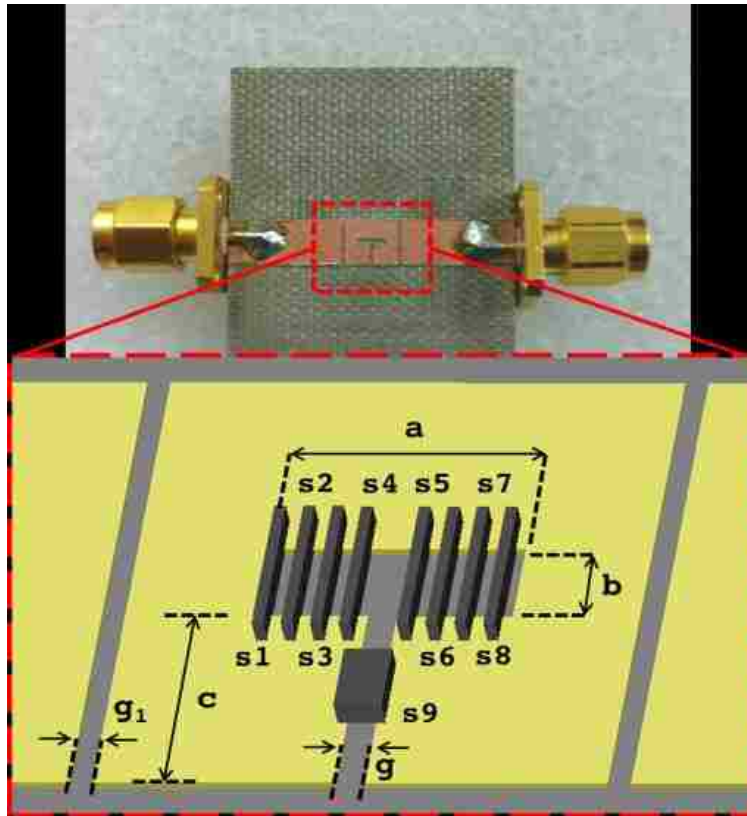


Fig. 4.26: Reconfigurable filter

The reconfigurability of the filter is achieved by integrating 9 switches within the T-slot. The switches are activated in pairs of two from the two edges of the T-slot. The purpose of the switches is to change the dimension of the slot (labeled ‘a’ in figure (4.26)) in order to produce a reconfigurable band pass filter.

Table 4.7: NN parameters of reconfigurable band pass filter

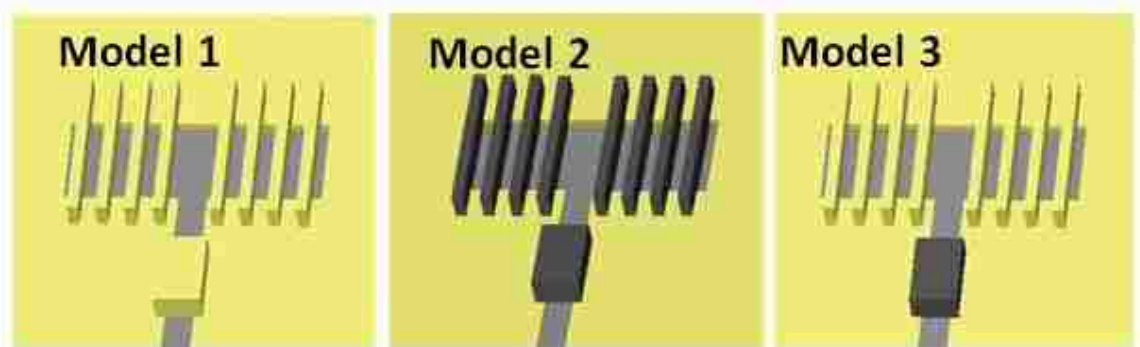
Iterations	15
Input Neurons	105
Output Neurons	9

Hidden Layers	1
Hidden Neurons	11

Training data for this filter are shown in appendix B, As shown in table (4.7), NN model requires 15 iterations to converge. The NN structure has 105 input neurons, 9 output neurons, and 11 hidden neurons in a single hidden layer.

The performance of the fabricated DMS filter was measured. Three different cases were taken:

- Model 1: All Switch ON
- Model 2: All Switch OFF
- Model 3: 8 Switches ON



Antenna response compared to NN for the three models are shown in Figures 4-27-29.

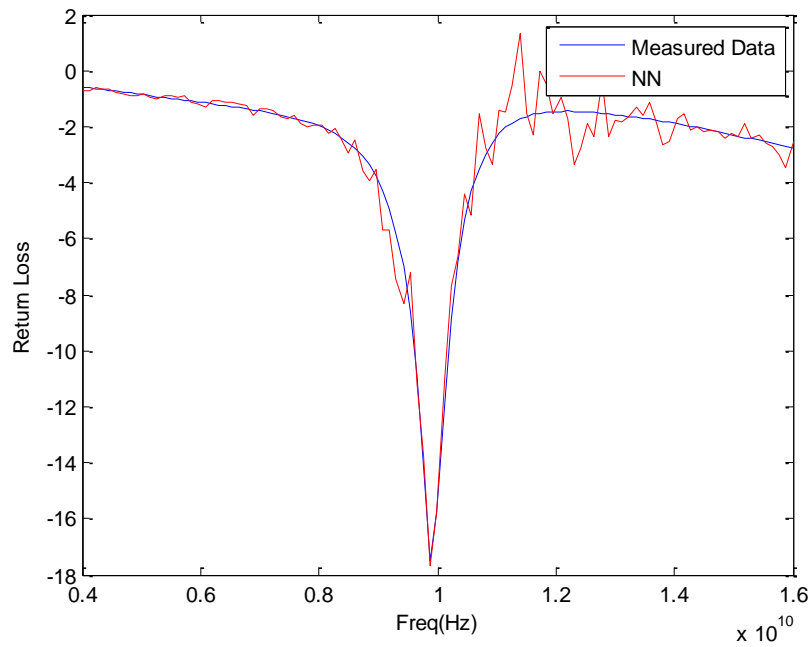


Fig. 4.27: NN output vs measured antenna response for Model 1

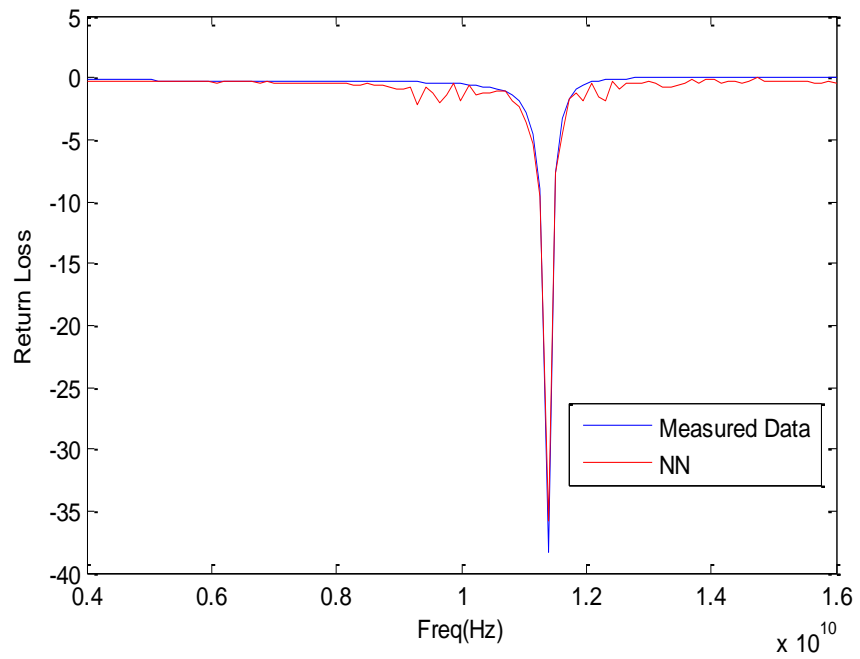


Fig. 4.28: NN output vs measured antenna response for Model 2

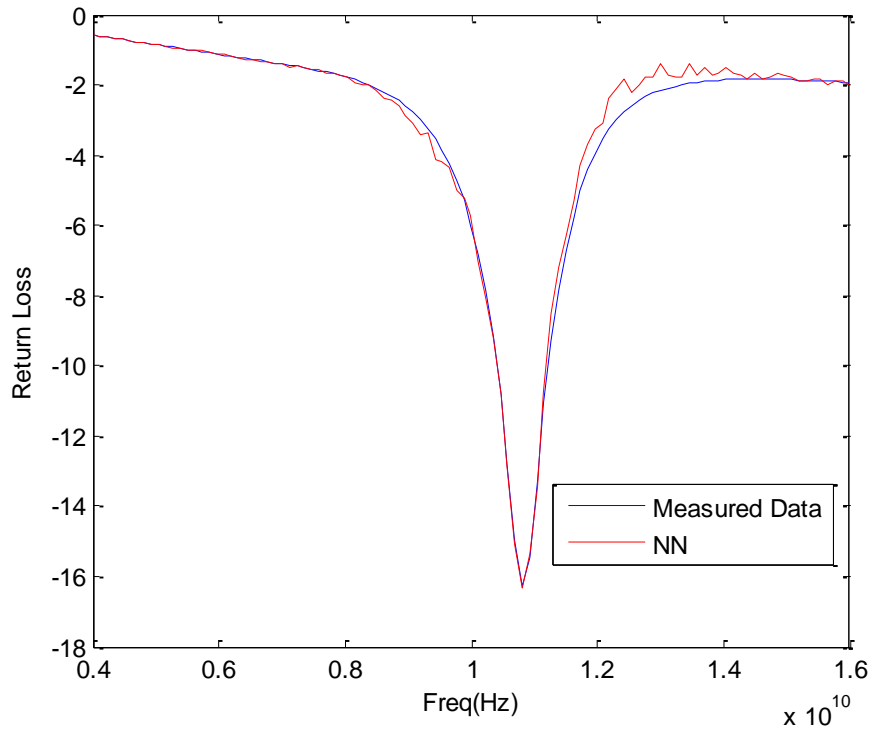


Fig. 4.29: NN output vs measured antenna response for Model 3

#### 4.6 Example 6: Star Antenna

In this example, a star shaped antenna [36] is modeled. The basic structure of the antenna, shown in figure (4.30), consists of 3 layers. The lower layer, which constitutes the ground plane, covers the entire hexagon shaped substrate with a side of 2.915 cm. The middle substrate, has a relative dielectric constant  $\epsilon_r=4.4$  and height 0.32cm. The upper layer, which is the patch, completely covers the hexagonal top surface. Six triangle slots of sides 1.2 cm and 1.4 cm and a base of 0.73 cm are cut out of the patch giving it the shape of a six armed star. In the star patch six rectangular slots of length 1.4 cm and width 0.2

cm were cut on each branch of the star as shown in figure (4.30). Six square switches of side 0.2 cm were mounted at the center of the rectangular slots. The antenna is fed by a coaxial probe with 50 input impedance.

When switches are activated, links between the center area of the star and the branches are established, which will change the surface current distribution and will create new current paths and new radiation edges, which give the antenna new resonances and different operation.

Training data for this antenna are shown in appendix B, Neural network model of this antenna has 51 input neurons, 11 hidden neurons, and 6 output neurons. The antenna structure is shown in figure (4.30), its NN structure is represented in figure (4.31).

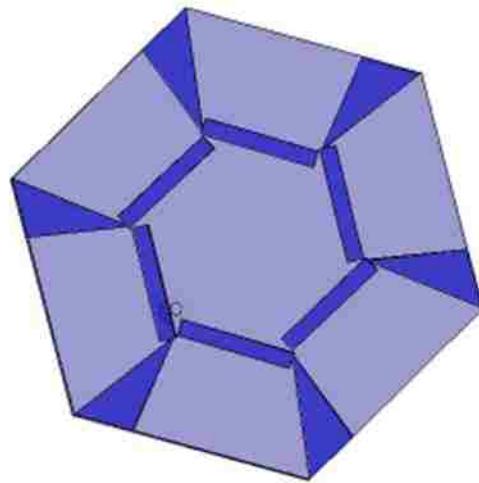


Fig. 4.30: Star antenna structure

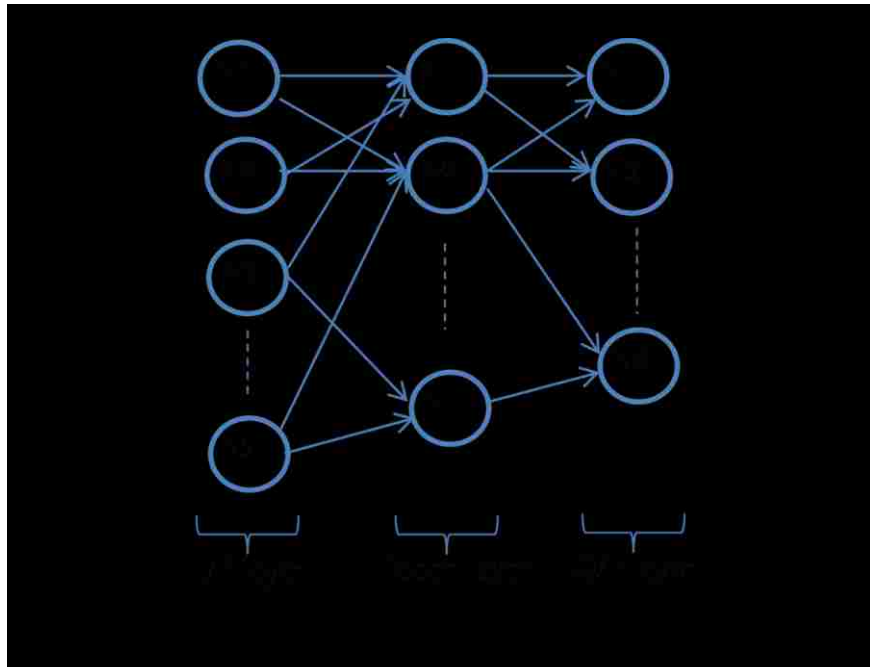


Fig. 4.31: NN model for Star antenna

Table 4.8: star antenna neural network parameters

Iterations	18
Input Neurons	51
Output Neurons	6
Hidden Layers	1
Hidden Neurons	11

As shown in table (4.8), neural network training requires 18 iterations to achieve the required accuracy with 51 input neurons and 6 output neurons.

Number of hidden neurons is found to be 11 as shown in figure (4.32).

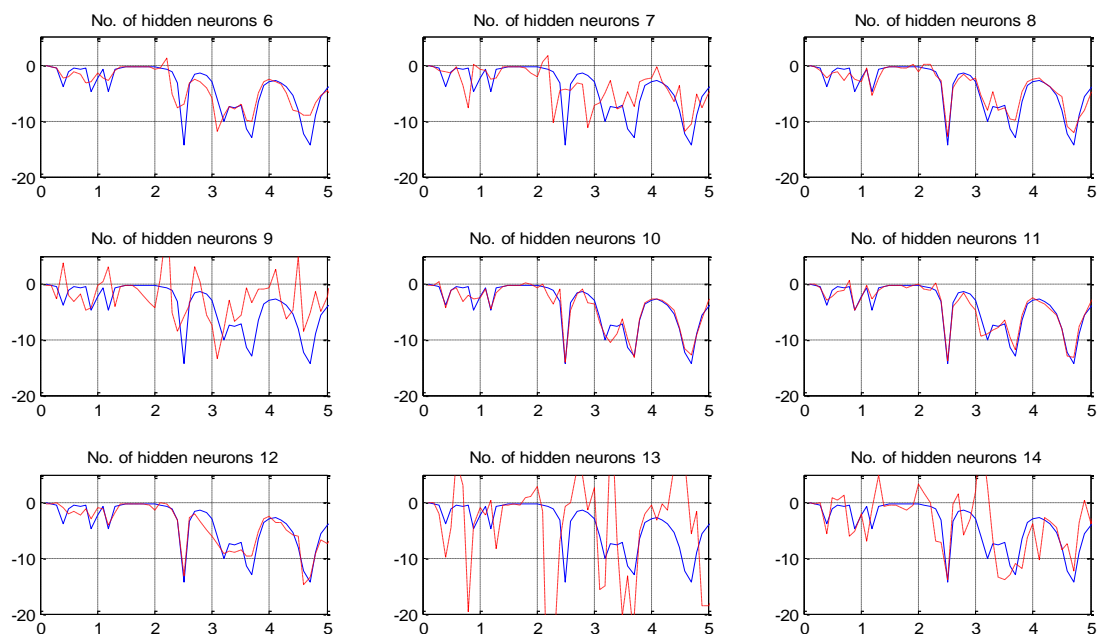


Fig. 4.32: Determining number of hidden neurons, NN output (red) vs measured antenna response (blue)

Figures (4.33) and (4.34) show the neural network output compared to the measured antenna response for some different switches configurations.

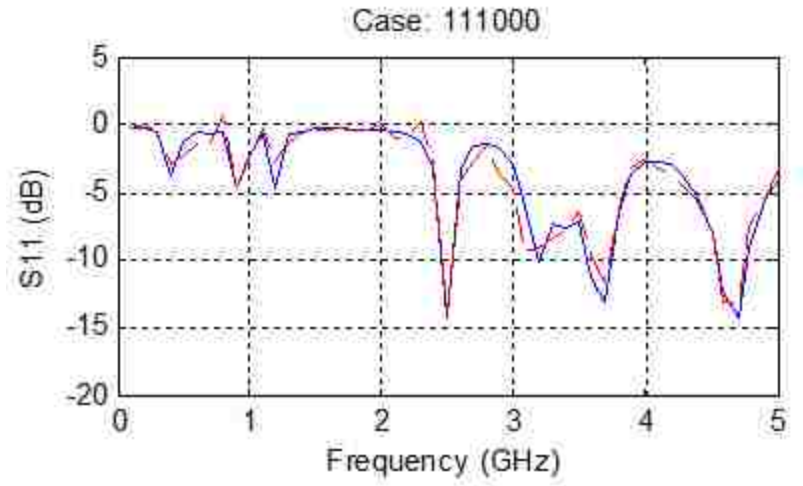
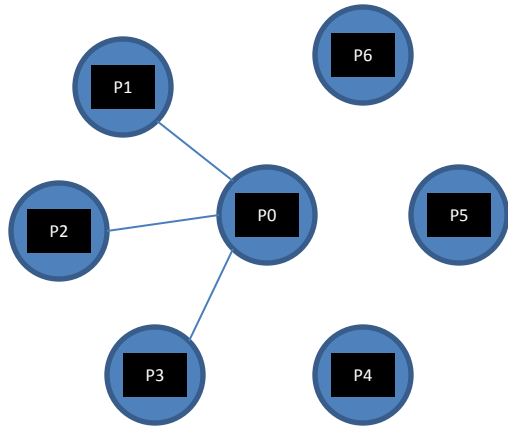


Fig. 4.33: NN output vs measured antenna response for case 111000

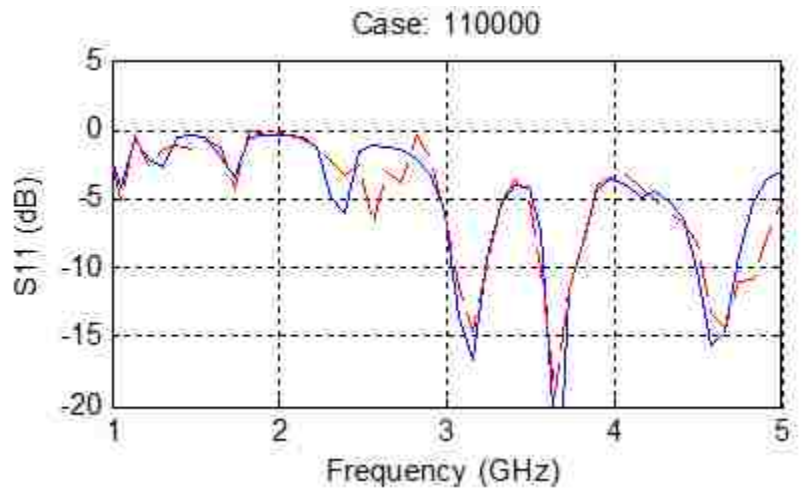
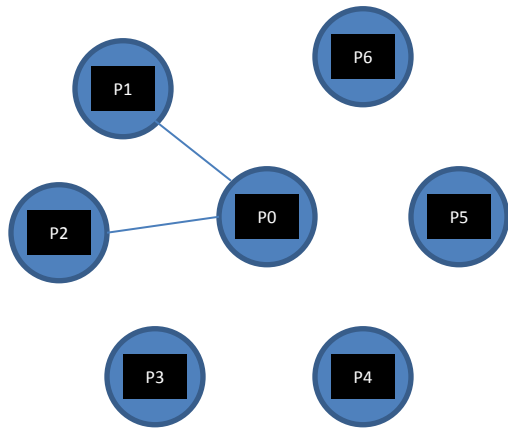


Fig. 4.34: NN output vs measured antenna response for case 110000



## CHAPTER V: CONCLUSIONS AND FUTURE WORK

In this work, a neural network model that can be embedded on an FPGA for reconfigurable antennas was described and designed.

Simulations from neural network models showed a great match with measured data; this is due to the distinguished characteristics of neural networks such as to learn from data, to generalize patterns in data and to model nonlinear relationships.

Neural network blocks were built in Matlab Simulink environment, and use basic Xilinx System Generator (XSG) blocks. Error caused by using XSG blocks was minimized by developing a new approach for approximating activation function.

An HDL code was generated using Xilinx System Generator and sent to an FPGA board using Xilinx ISE. Network parameters were mapped into a hardware structure that improves both performance and efficiency of the network. With a NN embedded on the FPGA board, we have a highly reconfigurable system in real time controller that thinks exactly as the system it models.

Several examples of reconfigurable antennas were given, showing the procedure of building the NN-FPGA controller that represents antenna parameters.

This work was a seed for a NN\_FPGA controller design. The design presented requires user to first build and train NN in Matlab environment. However, this does not use NN\_FPGA combination in its max. In future, NN training and building could be done inside the FPGA board itself; that is, a general NN code is sent to FPGA board, and then

the reconfigurable system is connected to the board, then NN will be programmed to take input/output pairs from the reconfigurable system. By this NN will be able to build and train itself by itself. This gives a highly reconfigurable controller in real time that is able to adapt with any changes in the system it controllers.

## APPENDIX A: MATLAB CODES

This appendix contains Matlab codes developed for each antenna NN model. Codes were written and run on Matlab 2009a.

### I.1 Filtenna antenna:

```
%
% Copyright (C) 2011, University of New Mexico
% All Rights Reserved by the University of New Mexico
% Author: Eyad Al Zuraiqi, ECE Departement, University of New Mexico.
% =====
% Filtenna Antenna
% This code generates a NN model for Filtenna antenna
% The code reads from xls file Filtenna_S.xls, and generates a 1:2:1
% (input neurons:hidden neurons:output neurons) back-propagation neural
% network with sigmoid activation function in the hidden layer.
%
% Antenna reference: Y. Tawk, J. Costantine, and C. G. Christodoulou,
%"A
% Varactor Based Reconfigurable Filtenna",IEEE Antennas and Wireless
% Propagation Letters.

clear all
close all
%clc
tic

% Read Antenna data
Datax = xlsread('Filtenna_S.xls');

% Input pattern
Antenna_Input = (Datax(:,1)/1e9)';

% Desired output
Antenna_Desired = Datax(:,2)';

% Generate the neural network
% Defaults
C = 0;
Hidden_Neurons = 2;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns','removeconstantrows','mapminmax'};
tpf = {'removeconstantrows','mapminmax'};
ddf = 'dividerand';

% Architecture
```

```

Nl = length(Hidden_Neurons)+1;
N_Net = network;
N_Net.numInputs = 1;
N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptFcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainFcn = btf;
N_Net.divideFcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;

```

```

N_Net.outputs{N1}.exampleOutput = Antenna_Desired;

% Initialize
N_Net = init(N_Net);

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```

## I.2 Microstrip Antenna

```
%
% Copyright (C) 2010, University of New Mexico
% All Rights Reserved by the University of New Mexico
% Author: Eyad Al Zuraiqi, ECE Departement, University of New Mexico.
% =====
% Reconfigurable multi-band stacked Microstrip Patch Antenna
% This code generates a NN model for the antenna
% The code generates a 15:8:2
% (input neurons:hidden neurons:output neurons) back-propagation neural
% network with sigmoid activation function in the hidden layer.
%
% Antenna referance: "Alayesh, M.A.; Christodoulou, C.G.; Joler, M.;
% Barbin, S.E.; , "Reconfigurable multi-band stacked Microstrip Patch
% Antenna for wireless applications," Antennas and Propagation
% Conference,
% 2008. LAPC 2008. Loughborough , vol., no., pp.329-332, 17-18 March
% 2008.

clear all
close all
%clc
tic

% Read Antenna data
Datax = xlsread('Mah.xls');

% Input pattern
Antenna_Input = Datax(:,2:5);

% Desired output
Antenna_Desired = [0 1 0 1; 0 0 1 1];

% Generate the neural network
% Defaults
C = 0;
Hidden_Neurons = 8;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns','removeconstantrows','mapminmax'};
tpf = {'removeconstantrows','mapminmax'};
ddf = 'dividerand';

% Architecture
Nl = length(Hidden_Neurons)+1;
N_Net = network;
N_Net.numInputs = 1;
N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
```

```

[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptfcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainfcn = btf;
N_Net.dividefcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;
N_Net.outputs{Nl}.exampleOutput = Antenna_Desired;

% Initialize
N_Net = init(N_Net);

```

```

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```



### I.3 Rotatable Microstrip Antenna

```
%
% Copyright (C) 2011, University of New Mexico
% All Rights Reserved by the University of New Mexico
% Author: Eyad Al Zuraiqi, ECE Departement, University of New Mexico.
% =====
% Rotatable Microstrip Antenna
% This code generates a NN model for Rotatable Microstrip Antenna
% The code reads from xls file Antenna1.xls, and generates a 201:7:4
% (input neurons:hidden neurons:output neurons) back-propagation neural
% network with sigmoid activation function in the hidden layer.
%
% Antenna reference: Tawk, Y.; Costantine, J.; Christodoulou, C.G.; ,
"A
% frequency reconfigurable rotatable microstrip antenna design,"
Antennas
% and Propagation Society International Symposium (APSURSI), 2010 IEEE
,
% vol., no., pp.1-4, 11-17 July 2010

clear all
close all
%clc
tic

% Read Antenna data
Datax = xlsread('Antenna1.xls');

% Input pattern
Antenna_Input = Datax(:,2:5);

% Desired output
S1 = [1 0 0 0];
S2 = [0 1 0 0];
S3 = [0 0 1 0];
S4 = [0 0 0 1];
Antenna_Desired = [S1;S2;S3;S4];

% Generate the neural network
% Defaults
C = 0;
Hidden_Neurons = 7;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns','removeconstantrows','mapminmax'};
tpf = {'removeconstantrows','mapminmax'};
ddf = 'dividerand';

% Architecture
N1 = length(Hidden_Neurons)+1;
N_Net = network;
N_Net.numInputs = 1;
```

```

N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptfcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainfcn = btf;
N_Net.dividefcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;
N_Net.outputs{Nl}.exampleOutput = Antenna_Desired;

% Initialize

```

```

N_Net = init(N_Net);

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```

## I.4 Planar Ultra-Wideband Antenna

```
%
% Copyright (C) 2011, University of New Mexico
% All Rights Reserved by the University of New Mexico
% Author: Eyad Al Zuraiqi, ECE Departement, University of New Mexico.
% =====
% Planar Ultra-Wideband Antenna
% This code generates a NN model for Planar Ultra-Wideband Antenna
% The code reads from xls file Antenna2.xls, and generates a 201:8:4
% (input neurons:hidden neurons:output neurons) back-propagation neural
% network with sigmoid activation function in the hidden layer.
%
% Antenna reference: Al-Husseini, M.; Ramadan, A.; Tawk, Y.;
Christodoulou,
% C.; Kabalan, K.; El-Hajj, A.; , "A planar ultrawideband antenna with
% multiple controllable band notches for UWB cognitive radio
applications,"
% Antennas and Propagation (EUCAP), Proceedings of the 5th European
% Conference on , vol., no., pp.375-377, 11-15 April 2011.

clear all
close all
%clc
tic

% Read Antenna data
Datax = xlsread('Antenna2.xls');

% Input pattern
Antenna_Input = Datax(:,2:9);

% Desired output
S1 = [0 1 1 0 0 1 0 1];
S2 = [1 0 1 0 1 0 0 1];
S3 = [0 0 1 0 1 1 1 0];
S4 = [0 0 1 0 1 1 1 0];
Antenna_Desired = [S1;S2;S3;S4];

% Generate the neural network
% Defaults
C = 0;
Hidden_Neurons = 8;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns','removeconstantrows','mapminmax'};
tpf = {'removeconstantrows','mapminmax'};
ddf = 'dividerand';

% Architecture
N1 = length(Hidden_Neurons)+1;
N_Net = network;
```

```

N_Net.numInputs = 1;
N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptfcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainfcn = btf;
N_Net.dividefcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;
N_Net.outputs{Nl}.exampleOutput = Antenna_Desired;

```

```

% Initialize
N_Net = init(N_Net);

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```

## I.5 Broadband Tapered Slot Antenna:

```
%
% Copyright (C) 2011, University of New Mexico
% All Rights Reserved by the University of New Mexico
% Author: Eyad Al Zuraiqi, ECE Departement, University of New Mexico.
% =====
% Broadband Tapered Slot Antenna
% This code generates a NN model for Broadband Tapered Slot Antenna
% The code reads from xls file Antenna3.xls, and generates a 105:11:9
% (input neurons:hidden neurons:output neurons) back-propagation neural
% network with sigmoid activation function in the hidden layer.
%
% Antenna reference: M. E. Zamudio, J. -H. Kim, Y. Tawk and C. G.
% Christodoulou, " Integrated Cognitive Radio Antenna Using
% Reconfigurable
% Band Pass Filters", European Conference on Antennas and Propagation
% 2011,
% 11-15 April 2011, Rome.

clear all
close all
clc
tic

% Read Antenna data
Datax = xlsread('Antenna3.xls');

% Input pattern
Antenna_Input = Datax(:,2:7);

% Desired output
S1 = [1 0 0 0 0 0];
S2 = [0 1 0 0 0 0];
S3 = [0 0 1 0 0 0];
S4 = [0 0 0 1 0 0];
S5 = [0 0 0 0 1 0];
S0 = [0 0 0 0 0 1]; % ALL OFF
Antenna_Desired = [S1;S2;S3;S4;S5;S0];

% Generate the neural network
% Defaults
C = 0;
Hidden_Neurons = 11;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns','removeconstantrows','mapminmax'};
tpf = {'removeconstantrows','mapminmax'};
ddf = 'dividerand';

% Architecture
N1 = length(Hidden_Neurons)+1;
N_Net = network;
```

```

N_Net.numInputs = 1;
N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptfcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainfcn = btf;
N_Net.dividefcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;
N_Net.outputs{Nl}.exampleOutput = Antenna_Desired;

```



```

% Initialize
N_Net = init(N_Net);

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```



```

C = 0;
Hidden_Neurons = 11;
Act_fun = {'logsig'};
btf = 'trainlm';
blf = 'learngdm';
pf = 'mse';
ipf = {'fixunknowns', 'removeconstantrows', 'mapminmax'};
tpf = {'removeconstantrows', 'mapminmax'};
ddf = 'dividerand';

% Architecture
Nl = length(Hidden_Neurons)+1;
N_Net = network;
N_Net.numInputs = 1;
N_Net.numLayers = Nl;
N_Net.biasConnect = ones(Nl,1);
N_Net.inputConnect(1,1) = 1;
[j,i] = meshgrid(1:Nl,1:Nl);
N_Net.layerConnect = (j == (i-1));
N_Net.outputConnect(Nl) = 1;

% Simulation
N_Net.inputs{1}.processFcns = ipf;
for i=1:Nl
    if (i < Nl)
        N_Net.layers{i}.size = Hidden_Neurons(i);
        if (Nl == 2)
            N_Net.layers{i}.name = 'Hidden Layer';
        else
            N_Net.layers{i}.name = ['Hidden Layer ' num2str(i)];
        end
    else
        N_Net.layers{i}.name = 'Output Layer';
    end
    if (length(Act_fun) < i) || all(isnan(Act_fun{i}))
        if (i<Nl)
            N_Net.layers{i}.transferFcn = 'tansig';
        else
            N_Net.layers{i}.transferFcn = 'purelin';
        end
    else
        N_Net.layers{i}.transferFcn = Act_fun{i};
    end
end
N_Net.outputs{Nl}.processFcns = tpf;

% Adaption
N_Net.adaptfcn = 'adaptwb';
N_Net.inputWeights{1,1}.learnFcn = blf;
for i=1:Nl
    N_Net.biases{i}.learnFcn = blf;
    N_Net.layerWeights{i,:}.learnFcn = blf;
end

% Training
N_Net.trainfcn = btf;

```

```

N_Net.divideFcn = ddf;
N_Net.performFcn = pf;

% Initialization
N_Net.initFcn = 'initlay';
for i=1:Nl
    N_Net.layers{i}.initFcn = 'initnw';
end

% Configuration
N_Net.inputs{1}.exampleInput = Antenna_Input;
N_Net.outputs{Nl}.exampleOutput = Antenna_Desired;

% Initialize
N_Net = init(N_Net);

% Plots
N_Net.plotFcns = {'plotperform','plottrainstate','plotregression'};

% NN simulation to Simulink model
y1 = sim(N_Net,Antenna_Input);

% Train the NN
[N_Net2,tr,Y,E,Xf,Af] = train(N_Net,Antenna_Input,Antenna_Desired);

% Generate a Simulink block diagram of the NN
gensim(N_Net2,-1)

% Test NN: Plot NN output vs antenna response
y2 = sim(N_Net2,Antenna_Input);
plot(Antenna_Input,Antenna_Desired,'o',Antenna_Input,y2,'*')
xlabel('Freq(GHz)')
ylabel('Return Loss')
legend('Measured Data','NN')

toc

```

## APPENDIX B: INPUT/OUTPUT PAIRS

This appendix contains data files for input/output pairs for antennas used for neural network modeling.

### B1. Filtenna Antenna

Freq.	S11 at voltage								
	11	13	15	17	19	21	23	25	27
6.10E+09	-2.4842	-2.4773	-3.4794	-2.4765	-1.4681	-1.4714	-1.4552	-1.4681	-1.4445
6.10E+09	-2.4506	-2.4545	-3.4599	-2.4374	-1.4385	-1.4432	-1.4308	-1.4409	-1.4274
6.11E+09	-2.4351	-2.4434	-3.4311	-2.4136	-1.4149	-1.4208	-1.406	-1.405	-1.414
6.11E+09	-2.415	-2.4189	-3.4108	-2.3895	-1.4042	-1.3963	-1.3738	-1.392	-1.3823
6.11E+09	-2.3996	-2.3894	-3.3929	-2.3745	-1.3947	-1.3774	-1.3728	-1.3801	-1.3623
6.11E+09	-2.3741	-2.3574	-3.3731	-2.3558	-1.3749	-1.3587	-1.3707	-1.3775	-1.359
6.12E+09	-2.3678	-2.3693	-3.3688	-2.3605	-1.3627	-1.3557	-1.3277	-1.3539	-1.3526
6.12E+09	-2.3419	-2.3413	-3.3262	-2.3197	-1.3102	-1.3208	-1.324	-1.3124	-1.3095
6.12E+09	-2.3367	-2.3365	-3.332	-2.3165	-1.315	-1.331	-1.3034	-1.3015	-1.315
6.12E+09	-2.3257	-2.3101	-3.3169	-2.2886	-1.3203	-1.3069	-1.2921	-1.2916	-1.2915
6.13E+09	-2.2897	-2.2753	-3.283	-2.2596	-1.2759	-1.2789	-1.2617	-1.2565	-1.2615
6.13E+09	-2.2789	-2.2678	-3.2682	-2.2558	-1.2565	-1.2597	-1.2446	-1.2409	-1.2477
6.13E+09	-2.2799	-2.2623	-3.268	-2.2571	-1.26	-1.2473	-1.2446	-1.2352	-1.246
6.13E+09	-2.2628	-2.2418	-3.2433	-2.2265	-1.2279	-1.2172	-1.2236	-1.2121	-1.2215
6.14E+09	-2.2422	-2.2188	-3.2176	-2.217	-1.2128	-1.2045	-1.1966	-1.2059	-1.199
6.14E+09	-2.2357	-2.2035	-3.2128	-2.1914	-1.2032	-1.1959	-1.198	-1.1744	-1.1836
6.14E+09	-2.2025	-2.183	-3.1894	-2.1514	-1.18	-1.1676	-1.1718	-1.1646	-1.1655
6.14E+09	-2.2009	-2.1804	-3.1733	-2.1644	-1.17	-1.1519	-1.1519	-1.1437	-1.1622
6.15E+09	-2.1797	-2.1565	-3.174	-2.1523	-1.1432	-1.1258	-1.1266	-1.1332	-1.1399
6.15E+09	-2.1894	-2.1407	-3.1501	-2.132	-1.1219	-1.129	-1.1563	-1.1325	-1.1219
6.15E+09	-2.1698	-2.1558	-3.1483	-2.1336	-1.1395	-1.1351	-1.1395	-1.1268	-1.1217
6.15E+09	-2.1835	-2.153	-3.1503	-2.1299	-1.1344	-1.1302	-1.1348	-1.1184	-1.1169
6.16E+09	-2.1741	-2.1507	-3.1403	-2.1334	-1.1334	-1.1196	-1.1356	-1.1293	-1.1166
6.16E+09	-2.158	-2.1326	-3.1302	-2.1175	-1.1181	-1.1121	-1.1181	-1.1066	-1.1041
6.16E+09	-2.1539	-2.1281	-3.1308	-2.1185	-1.1133	-1.1084	-1.1126	-1.1017	-1.0939
6.16E+09	-2.1386	-2.1144	-3.1165	-2.1093	-1.1007	-1.0947	-1.0985	-1.0849	-1.0753
6.17E+09	-2.1487	-2.1239	-3.1165	-2.1122	-1.0935	-1.0932	-1.0971	-1.0871	-1.0767

6.17E+09	-2.1275	-2.111	-3.0922	-2.0898	-1.0707	-1.0769	-1.0764	-1.0678	-1.0471
6.17E+09	-2.1475	-2.108	-3.0887	-2.0923	-1.0715	-1.0799	-1.0657	-1.0614	-1.0535
6.17E+09	-2.1125	-2.0993	-3.0972	-2.108	-1.0895	-1.078	-1.0854	-1.0666	-1.0548
6.18E+09	-2.1381	-2.1028	-3.0948	-2.0969	-1.0681	-1.071	-1.071	-1.043	-1.0462
6.18E+09	-2.1274	-2.1221	-3.0697	-2.0777	-1.0599	-1.0655	-1.0508	-1.0592	-1.0401
6.18E+09	-2.1127	-2.0813	-3.0706	-2.0689	-1.0407	-1.0411	-1.0517	-1.0282	-1.0177
6.18E+09	-2.1003	-2.0829	-3.0726	-2.0783	-1.0469	-1.0612	-1.0357	-1.0356	-1.0382
6.19E+09	-2.0986	-2.0822	-3.0685	-2.0788	-1.0491	-1.047	-1.0362	-1.0386	-1.0146
6.19E+09	-2.1035	-2.0791	-3.0704	-2.0646	-1.0529	-1.0423	-1.037	-1.0209	-1.0278
6.19E+09	-2.1031	-2.0886	-3.0777	-2.0695	-1.0608	-1.0549	-1.0404	-1.0354	-1.0254
6.19E+09	-2.1031	-2.0802	-3.0699	-2.0631	-1.0578	-1.0503	-1.0289	-1.027	-1.0288
6.20E+09	-2.0875	-2.0688	-3.0499	-2.0451	-1.0264	-1.0284	-1.0131	-1.01	-1.0174
6.20E+09	-2.0947	-2.0818	-3.0732	-2.0559	-1.0658	-1.0482	-1.0301	-1.0266	-1.0292
6.20E+09	-2.1117	-2.0929	-3.0912	-2.0688	-1.0629	-1.0479	-1.0412	-1.0388	-1.0345
6.20E+09	-2.1428	-2.1323	-3.1222	-2.0926	-1.096	-1.0758	-1.0576	-1.0534	-1.064
6.21E+09	-2.1611	-2.1389	-3.1282	-2.1118	-1.1083	-1.1032	-1.0736	-1.0751	-1.0811
6.21E+09	-2.1668	-2.139	-3.1473	-2.1249	-1.1167	-1.1053	-1.1176	-1.1092	-1.096
6.21E+09	-2.1774	-2.1646	-3.1514	-2.1439	-1.1319	-1.1321	-1.1186	-1.1146	-1.1096
6.21E+09	-2.2203	-2.1845	-3.1668	-2.1703	-1.1443	-1.1312	-1.1329	-1.1035	-1.1203
6.22E+09	-2.2205	-2.209	-3.2086	-2.1942	-1.1764	-1.169	-1.1512	-1.1503	-1.1358
6.22E+09	-2.2424	-2.2235	-3.2078	-2.1938	-1.1767	-1.1662	-1.1576	-1.1441	-1.1443
6.22E+09	-2.2735	-2.2415	-3.2322	-2.2197	-1.215	-1.1967	-1.184	-1.1813	-1.1876
6.22E+09	-2.2965	-2.2814	-3.2618	-2.2505	-1.2437	-1.2276	-1.2157	-1.2057	-1.2164
6.23E+09	-2.3187	-2.2918	-3.2841	-2.2626	-1.2587	-1.2409	-1.2279	-1.227	-1.2231
6.23E+09	-2.3396	-2.3275	-3.3167	-2.2813	-1.2789	-1.2627	-1.2474	-1.2451	-1.2412
6.23E+09	-2.3671	-2.3526	-3.3183	-2.3129	-1.301	-1.2876	-1.2822	-1.2649	-1.2614
6.23E+09	-2.3881	-2.3666	-3.3383	-2.3214	-1.3094	-1.2985	-1.2838	-1.2911	-1.2731
6.24E+09	-2.4169	-2.4047	-3.3665	-2.3483	-1.3285	-1.3233	-1.3035	-1.2978	-1.2892
6.24E+09	-2.462	-2.4445	-3.4038	-2.3884	-1.3905	-1.367	-1.346	-1.3421	-1.3419
6.24E+09	-2.4877	-2.4729	-3.4412	-2.4265	-1.401	-1.3984	-1.3832	-1.3611	-1.3648
6.24E+09	-2.519	-2.506	-3.4733	-2.4519	-1.4507	-1.4319	-1.4052	-1.4076	-1.3897
6.25E+09	-2.5385	-2.5157	-3.4999	-2.4712	-1.4519	-1.4438	-1.4331	-1.4268	-1.4021
6.25E+09	-2.5575	-2.5504	-3.5191	-2.4919	-1.4806	-1.455	-1.4563	-1.4418	-1.4299
6.25E+09	-2.5745	-2.5474	-3.5249	-2.5015	-1.4806	-1.4604	-1.4603	-1.442	-1.4293
6.25E+09	-2.6062	-2.586	-3.5445	-2.5285	-1.5167	-1.5003	-1.4902	-1.4686	-1.4525
6.26E+09	-2.6458	-2.6242	-3.5811	-2.5573	-1.5441	-1.5244	-1.5097	-1.5016	-1.4904
6.26E+09	-2.6932	-2.6793	-3.6319	-2.6093	-1.588	-1.5705	-1.554	-1.5409	-1.527
6.26E+09	-2.7385	-2.7151	-3.666	-2.649	-1.6177	-1.6064	-1.5899	-1.5745	-1.561
6.26E+09	-2.7751	-2.7663	-3.6966	-2.6855	-1.6436	-1.6361	-1.6189	-1.6082	-1.5903
6.27E+09	-2.8122	-2.7967	-3.7389	-2.7169	-1.6839	-1.6575	-1.6552	-1.634	-1.6274
6.27E+09	-2.8611	-2.8283	-3.7966	-2.763	-1.7324	-1.6986	-1.6913	-1.6832	-1.6778

6.27E+09	-2.9114	-2.8894	-3.8115	-2.8089	-1.782	-1.7417	-1.7349	-1.7158	-1.7125
6.27E+09	-2.9537	-2.9321	-3.8741	-2.8411	-1.8014	-1.7835	-1.7531	-1.7294	-1.7278
6.28E+09	-3.0074	-2.9527	-3.8956	-2.8547	-1.8257	-1.8093	-1.7785	-1.7801	-1.7712
6.28E+09	-3.0507	-3.0351	-3.96	-2.9037	-1.8936	-1.8655	-1.8498	-1.832	-1.8293
6.28E+09	-3.1079	-3.0523	-4.0043	-2.9504	-1.9174	-1.8845	-1.8734	-1.8559	-1.8541
6.28E+09	-3.1585	-3.0958	-4.0439	-2.9941	-1.9572	-1.9321	-1.9116	-1.9014	-1.8813
6.29E+09	-3.2094	-3.1438	-4.0736	-3.0248	-1.9963	-1.9645	-1.95	-1.9236	-1.9249
6.29E+09	-3.27	-3.1942	-4.12	-3.0703	-2.0325	-2.0041	-1.9816	-1.9602	-1.9596
6.29E+09	-3.3217	-3.237	-4.171	-3.1198	-2.0804	-2.0448	-2.026	-2.0026	-2.0038
6.29E+09	-3.3754	-3.2832	-4.2138	-3.1634	-2.1134	-2.0815	-2.0573	-2.0404	-2.0246
6.30E+09	-3.4151	-3.322	-4.2454	-3.1867	-2.1492	-2.1178	-2.087	-2.0676	-2.055
6.30E+09	-3.5025	-3.4066	-4.3123	-3.2508	-2.2141	-2.1757	-2.1444	-2.1281	-2.1248
6.30E+09	-3.5594	-3.4702	-4.3719	-3.3139	-2.2708	-2.2407	-2.2104	-2.1846	-2.176
6.30E+09	-3.6308	-3.524	-4.4328	-3.3575	-2.3302	-2.2813	-2.2489	-2.2323	-2.2179
6.31E+09	-3.725	-3.5887	-4.5033	-3.435	-2.3829	-2.3484	-2.3105	-2.2841	-2.268
6.31E+09	-3.7764	-3.656	-4.5681	-3.487	-2.4274	-2.4079	-2.3674	-2.3459	-2.3189
6.31E+09	-3.8542	-3.7203	-4.6185	-3.5467	-2.4801	-2.4413	-2.3973	-2.3711	-2.3508
6.31E+09	-3.9067	-3.7785	-4.6584	-3.5885	-2.5245	-2.4728	-2.4492	-2.4193	-2.3918
6.32E+09	-3.9972	-3.8248	-4.718	-3.6402	-2.5843	-2.5254	-2.4978	-2.4617	-2.4342
6.32E+09	-4.0619	-3.8993	-4.787	-3.6987	-2.6394	-2.5714	-2.5252	-2.5113	-2.4801
6.32E+09	-4.187	-4.0111	-4.89	-3.7888	-2.7241	-2.6668	-2.6205	-2.6008	-2.5599
6.32E+09	-4.2995	-4.1088	-4.9826	-3.8745	-2.8014	-2.7344	-2.6957	-2.6584	-2.6269
6.33E+09	-4.4018	-4.2011	-5.0601	-3.9483	-2.8755	-2.8046	-2.76	-2.7287	-2.6907
6.33E+09	-4.4985	-4.2634	-5.131	-4.0123	-2.9362	-2.868	-2.8134	-2.7869	-2.7362
6.33E+09	-4.5951	-4.3599	-5.2084	-4.0765	-2.9894	-2.9166	-2.8664	-2.833	-2.7846
6.33E+09	-4.695	-4.4255	-5.282	-4.1335	-3.0455	-2.9646	-2.9092	-2.8827	-2.8417
6.34E+09	-4.8389	-4.5442	-5.3743	-4.2091	-3.122	-3.0352	-2.9866	-2.9341	-2.9004
6.34E+09	-4.9843	-4.6747	-5.4739	-4.3196	-3.2099	-3.1273	-3.0506	-2.9857	-2.9856
6.34E+09	-5.1429	-4.7952	-5.5805	-4.4009	-3.3019	-3.2153	-3.119	-3.0682	-3.06
6.34E+09	-5.288	-4.8843	-5.6806	-4.4789	-3.361	-3.2638	-3.1839	-3.1308	-3.1083
6.35E+09	-5.4305	-4.9865	-5.749	-4.5509	-3.4166	-3.3144	-3.2406	-3.1797	-3.1643
6.35E+09	-5.6275	-5.1275	-5.8861	-4.6646	-3.5174	-3.4094	-3.3195	-3.2677	-3.2404
6.35E+09	-5.807	-5.2442	-5.9692	-4.7356	-3.5857	-3.4744	-3.3646	-3.3192	-3.2824
6.35E+09	-6.0243	-5.4003	-6.1016	-4.8372	-3.6712	-3.5447	-3.449	-3.3838	-3.3542
6.36E+09	-6.2762	-5.5791	-6.2273	-4.9451	-3.7576	-3.6309	-3.5277	-3.4643	-3.4329
6.36E+09	-6.5715	-5.7806	-6.3782	-5.0699	-3.8718	-3.7226	-3.6174	-3.5526	-3.5127
6.36E+09	-6.8635	-5.977	-6.5358	-5.1816	-3.9631	-3.8076	-3.7003	-3.6346	-3.5892
6.36E+09	-7.1923	-6.1842	-6.688	-5.3113	-4.0684	-3.9034	-3.7973	-3.7012	-3.6665
6.37E+09	-7.5263	-6.3951	-6.8335	-5.4182	-4.1611	-3.9872	-3.8666	-3.7834	-3.7354
6.37E+09	-7.9237	-6.6122	-6.9972	-5.5599	-4.2567	-4.064	-3.9505	-3.8242	-3.8013
6.37E+09	-8.3745	-6.8933	-7.1832	-5.6858	-4.3465	-4.1646	-4.0241	-3.91	-3.8633

6.37E+09	-8.8883	-7.2018	-7.3857	-5.8342	-4.4706	-4.2496	-4.1079	-3.9786	-3.9323
6.38E+09	-9.4569	-7.5937	-7.6257	-6.0113	-4.6179	-4.3633	-4.2278	-4.0776	-4.0211
6.38E+09	-10.1156	-7.9905	-7.8954	-6.2003	-4.7699	-4.4958	-4.3372	-4.1966	-4.1095
6.38E+09	-10.7463	-8.4463	-8.19	-6.404	-4.9191	-4.627	-4.4599	-4.2834	-4.2119
6.38E+09	-11.2902	-8.8817	-8.4656	-6.5811	-5.0614	-4.7268	-4.5526	-4.3789	-4.2907
6.39E+09	-11.7953	-9.4459	-8.7844	-6.7969	-5.2174	-4.8283	-4.6481	-4.4585	-4.3581
6.39E+09	-12.0981	-10.1175	-9.1982	-7.0673	-5.4167	-4.9901	-4.78	-4.5778	-4.4553
6.39E+09	-12.0039	-10.806	-9.6507	-7.3427	-5.6162	-5.1409	-4.9039	-4.6948	-4.5609
6.39E+09	-11.5258	-11.493	-10.163	-7.6578	-5.828	-5.3093	-5.0305	-4.8	-4.655
6.40E+09	-10.7706	-12.1185	-10.6967	-7.9708	-6.0552	-5.4719	-5.1506	-4.9035	-4.7537
6.40E+09	-9.9113	-12.5903	-11.3306	-8.3558	-6.284	-5.6209	-5.2815	-5.0114	-4.8404
6.40E+09	-9.0142	-12.7376	-11.9837	-8.7685	-6.5669	-5.7974	-5.4216	-5.1335	-4.9399
6.40E+09	-8.2384	-12.5739	-12.7151	-9.224	-6.8109	-5.9845	-5.5303	-5.2423	-5.0468
6.41E+09	-7.5355	-12.0458	-13.4705	-9.7548	-7.11	-6.1728	-5.6611	-5.3467	-5.1453
6.41E+09	-6.9288	-11.2323	-14.2191	-10.4056	-7.5166	-6.4266	-5.8185	-5.4901	-5.2621
6.41E+09	-6.4084	-10.2709	-14.6652	-11.21	-8.0187	-6.7429	-6.0595	-5.6949	-5.4406
6.41E+09	-5.9685	-9.3151	-14.5375	-12.0884	-8.5955	-7.1069	-6.3195	-5.8945	-5.6186
6.42E+09	-5.6427	-8.4654	-13.9267	-12.9374	-9.185	-7.4453	-6.5634	-6.0976	-5.7854
6.42E+09	-5.3754	-7.7348	-13.0201	-13.7743	-9.8575	-7.8483	-6.8413	-6.2969	-5.9504
6.42E+09	-5.1614	-7.115	-11.9843	-14.3841	-10.6716	-8.3398	-7.1644	-6.5428	-6.1547
6.42E+09	-4.9971	-6.5954	-10.9646	-14.5605	-11.6029	-8.9138	-7.5383	-6.8167	-6.3834
6.43E+09	-4.8451	-6.1493	-9.9905	-14.09	-12.6146	-9.5824	-7.9626	-7.1229	-6.6285
6.43E+09	-4.7449	-5.7993	-9.1798	-13.145	-13.546	-10.3474	-8.4371	-7.4615	-6.9148
6.43E+09	-4.6433	-5.5144	-8.4818	-11.9904	-14.258	-11.1885	-8.9876	-7.8268	-7.1702
6.43E+09	-4.5762	-5.3053	-7.9272	-10.8648	-14.4031	-12.1451	-9.6408	-8.2484	-7.4887
6.44E+09	-4.5326	-5.1233	-7.4689	-9.8385	-13.962	-13.2323	-10.3782	-8.7308	-7.8402
6.44E+09	-4.4738	-4.9729	-7.082	-8.9065	-12.9603	-14.3263	-11.266	-9.3105	-8.2835
6.44E+09	-4.4384	-4.8446	-6.7716	-8.1367	-11.7463	-15.1484	-12.2229	-9.9601	-8.7543
6.44E+09	-4.4095	-4.7585	-6.4997	-7.4713	-10.4988	-15.5523	-13.4436	-10.7702	-9.3303
6.45E+09	-4.3915	-4.6495	-6.2877	-6.9102	-9.3182	-14.9018	-14.7155	-11.68	-10.0075
6.45E+09	-4.3788	-4.6088	-6.1199	-6.4714	-8.3266	-13.7465	-15.8885	-12.7937	-10.8176
6.45E+09	-4.3639	-4.5428	-5.9918	-6.1005	-7.4903	-12.3239	-16.3904	-14.0436	-11.7637
6.45E+09	-4.3705	-4.5093	-5.8843	-5.8102	-6.8309	-11.0578	-16.0873	-15.2816	-12.8008
6.46E+09	-4.3637	-4.4918	-5.7994	-5.5712	-6.267	-9.8911	-15.0486	-16.599	-14.109
6.46E+09	-4.3897	-4.4914	-5.7514	-5.3796	-5.7748	-8.8325	-13.6006	-17.4249	-15.6436
6.46E+09	-4.3956	-4.4752	-5.6846	-5.203	-5.3563	-7.9002	-11.9915	-17.0908	-17.2589
6.46E+09	-4.4159	-4.4691	-5.6511	-5.0781	-5.0272	-7.139	-10.6079	-15.5807	-18.1084
6.47E+09	-4.4352	-4.4828	-5.6284	-4.9882	-4.7799	-6.5339	-9.473	-13.9731	-17.8726
6.47E+09	-4.4532	-4.4938	-5.6056	-4.8935	-4.5556	-6.0321	-8.489	-12.4254	-16.666
6.47E+09	-4.4867	-4.5031	-5.5959	-4.8378	-4.3808	-5.617	-7.6781	-11.0109	-14.9235
6.47E+09	-4.5116	-4.5276	-5.6	-4.811	-4.2514	-5.2613	-6.9966	-9.8258	-13.2095



6.48E+09	-4.5401	-4.55	-5.6145	-4.762	-4.1296	-4.9768	-6.4193	-8.7658	-11.6556
6.48E+09	-4.5828	-4.5887	-5.6105	-4.7449	-4.0565	-4.7473	-5.9202	-7.9029	-10.301
6.48E+09	-4.6084	-4.6083	-5.6226	-4.7089	-3.9567	-4.5375	-5.498	-7.1451	-9.1382
6.48E+09	-4.646	-4.6417	-5.6495	-4.7177	-3.9149	-4.4031	-5.213	-6.5764	-8.2469
6.49E+09	-4.6892	-4.6866	-5.6823	-4.7291	-3.8916	-4.292	-4.9791	-6.1176	-7.5212
6.49E+09	-4.7204	-4.7123	-5.6956	-4.7431	-3.8672	-4.1995	-4.7762	-5.7287	-6.9145
6.49E+09	-4.7462	-4.7273	-5.7006	-4.734	-3.835	-4.1093	-4.5867	-5.3757	-6.3674
6.49E+09	-4.776	-4.7484	-5.7261	-4.7394	-3.8222	-4.0383	-4.4339	-5.087	-5.9088
6.50E+09	-4.8072	-4.783	-5.7497	-4.7536	-3.813	-3.9928	-4.3113	-4.8491	-5.5226
6.50E+09	-4.8284	-4.7991	-5.7606	-4.7709	-3.8004	-3.9406	-4.2161	-4.6694	-5.2227
6.50E+09	-4.8602	-4.8302	-5.7796	-4.7868	-3.8018	-3.9105	-4.1426	-4.5025	-4.9783
6.50E+09	-4.8946	-4.8584	-5.8125	-4.7963	-3.808	-3.8913	-4.0722	-4.3799	-4.7889
6.51E+09	-4.9433	-4.8849	-5.8293	-4.8198	-3.8266	-3.8898	-4.0429	-4.3037	-4.6269
6.51E+09	-4.9721	-4.9196	-5.8781	-4.8566	-3.8584	-3.8934	-4.0298	-4.2219	-4.4843
6.51E+09	-5.0246	-4.9782	-5.9237	-4.8934	-3.885	-3.9077	-4.0237	-4.1848	-4.4063
6.51E+09	-5.0384	-4.9868	-5.9443	-4.8967	-3.8961	-3.8997	-3.9938	-4.1315	-4.3139
6.52E+09	-5.097	-5.0431	-5.9773	-4.945	-3.923	-3.9306	-3.999	-4.0986	-4.2555
6.52E+09	-5.1488	-5.094	-6.0384	-4.988	-3.972	-3.9649	-4.0057	-4.0921	-4.2129
6.52E+09	-5.1744	-5.12	-6.0693	-5.0173	-3.9916	-3.9786	-4.0064	-4.0666	-4.1755
6.52E+09	-5.2192	-5.1631	-6.1157	-5.0592	-4.0302	-4.0162	-4.0357	-4.087	-4.1701
6.53E+09	-5.2629	-5.2107	-6.1537	-5.102	-4.0762	-4.0538	-4.0575	-4.1053	-4.1619
6.53E+09	-5.3021	-5.2493	-6.2113	-5.1322	-4.1158	-4.0942	-4.0884	-4.1181	-4.16
6.53E+09	-5.3296	-5.268	-6.2127	-5.1616	-4.1279	-4.0845	-4.0798	-4.1014	-4.1336
6.53E+09	-5.3428	-5.2548	-6.2315	-5.1911	-4.1715	-4.1149	-4.0993	-4.1118	-4.1275
6.54E+09	-5.3973	-5.3012	-6.2797	-5.2381	-4.2126	-4.1531	-4.1372	-4.14	-4.1467
6.54E+09	-5.4078	-5.3497	-6.3161	-5.2614	-4.2157	-4.183	-4.1521	-4.1545	-4.1478
6.54E+09	-5.4531	-5.3874	-6.3325	-5.2942	-4.2671	-4.2317	-4.1789	-4.1852	-4.169
6.54E+09	-5.4734	-5.4183	-6.3816	-5.3271	-4.2986	-4.2553	-4.2069	-4.2133	-4.1856
6.55E+09	-5.5033	-5.4554	-6.4096	-5.3606	-4.3545	-4.2849	-4.225	-4.2208	-4.2146
6.55E+09	-5.5406	-5.4895	-6.4442	-5.4065	-4.3857	-4.3307	-4.2609	-4.2602	-4.2187
6.55E+09	-5.5762	-5.5233	-6.4731	-5.452	-4.4174	-4.3595	-4.3036	-4.2995	-4.2547
6.55E+09	-5.6001	-5.5674	-6.5166	-5.5027	-4.4558	-4.4023	-4.3404	-4.334	-4.2859
6.56E+09	-5.6056	-5.5899	-6.5517	-5.5228	-4.4735	-4.4353	-4.3676	-4.3556	-4.31
6.56E+09	-5.6342	-5.6125	-6.5677	-5.5583	-4.5168	-4.4684	-4.3941	-4.393	-4.3402
6.56E+09	-5.6732	-5.6604	-6.6279	-5.6108	-4.5566	-4.5074	-4.4413	-4.4266	-4.3834
6.56E+09	-5.7134	-5.6983	-6.6862	-5.6383	-4.5884	-4.5671	-4.4923	-4.4728	-4.4272
6.57E+09	-5.7893	-5.7634	-6.7272	-5.7102	-4.6508	-4.6047	-4.5451	-4.5196	-4.4858
6.57E+09	-5.8087	-5.8173	-6.7851	-5.7512	-4.667	-4.6711	-4.6027	-4.5552	-4.5265
6.57E+09	-5.8365	-5.8495	-6.8341	-5.772	-4.7065	-4.6717	-4.6347	-4.5766	-4.5772
6.57E+09	-5.8869	-5.8881	-6.8581	-5.8034	-4.7206	-4.7087	-4.6506	-4.5988	-4.616
6.58E+09	-5.9474	-5.9389	-6.9133	-5.8492	-4.7872	-4.767	-4.7114	-4.6395	-4.6747

6.58E+09	-6.0111	-5.9983	-6.9688	-5.9094	-4.8419	-4.8197	-4.7638	-4.7102	-4.7236
6.58E+09	-6.0487	-6.0405	-7.0127	-5.9453	-4.8714	-4.839	-4.7998	-4.7459	-4.7622
6.58E+09	-6.0795	-6.0661	-7.0322	-5.9566	-4.8712	-4.852	-4.827	-4.7617	-4.782
6.59E+09	-6.1243	-6.0987	-7.0609	-5.9923	-4.9049	-4.8756	-4.8563	-4.7906	-4.8121
6.59E+09	-6.1376	-6.1063	-7.0601	-6.011	-4.9265	-4.8964	-4.8646	-4.8114	-4.8226
6.59E+09	-6.1519	-6.129	-7.0692	-6.0172	-4.9557	-4.9098	-4.8923	-4.8292	-4.8331
6.59E+09	-6.1805	-6.1631	-7.1108	-6.0407	-4.9779	-4.9332	-4.9194	-4.86	-4.8671
6.60E+09	-6.2554	-6.2235	-7.1431	-6.0853	-5.0244	-4.9698	-4.9873	-4.9131	-4.9216
6.60E+09	-6.2883	-6.2419	-7.1689	-6.1197	-5.0526	-4.9919	-5.0142	-4.9409	-4.9232
6.60E+09	-6.3045	-6.2675	-7.1944	-6.1398	-5.0902	-5.0386	-5.0434	-4.9632	-4.9545

## B2. Rotatable Microstrip Antenna

Freq.	S11			
	Shape 1	Shape 2	Shape 3	Shape 4
1E+09	-3.60107	-1.91272	-0.81233	-2.56551
1.03E+09	-3.89904	-2.02405	-0.86851	-2.59963
1.06E+09	-4.3262	-2.13951	-0.95373	-2.67448
1.09E+09	-4.51888	-2.1646	-1.02678	-2.78004
1.12E+09	-4.38077	-1.96888	-1.08648	-2.68767
1.15E+09	-4.25468	-1.90645	-1.15717	-2.58474
1.18E+09	-3.79683	-1.8556	-1.20632	-2.50412
1.21E+09	-3.58195	-1.9033	-1.19548	-2.49433
1.24E+09	-3.45818	-2.08991	-1.25395	-2.52675
1.27E+09	-3.33157	-2.25347	-1.28642	-2.46562
1.3E+09	-3.21165	-2.30279	-1.34839	-2.43423
1.33E+09	-3.05674	-2.33722	-1.36757	-2.42553
1.36E+09	-2.99656	-2.40222	-1.36659	-2.39273
1.39E+09	-2.92427	-2.35351	-1.40076	-2.34526
1.42E+09	-2.86565	-2.35209	-1.45054	-2.30741
1.45E+09	-2.80861	-2.63707	-1.53377	-2.33817
1.48E+09	-2.68421	-2.74668	-1.60795	-2.31579
1.51E+09	-2.59547	-2.83273	-1.68005	-2.30048
1.54E+09	-2.50083	-2.85747	-1.75478	-2.32636
1.57E+09	-2.39243	-3.04945	-1.77312	-2.27821
1.6E+09	-2.36023	-3.16739	-1.82508	-2.16638
1.63E+09	-2.31946	-3.30383	-1.87498	-2.09631
1.66E+09	-2.33039	-3.59362	-1.92028	-2.09903
1.69E+09	-2.32108	-3.69979	-1.95541	-2.11291
1.72E+09	-2.35537	-3.74626	-1.93314	-2.08399
1.75E+09	-2.38356	-4.01493	-1.9068	-2.08641
1.78E+09	-2.40455	-4.09587	-1.91865	-2.13649
1.81E+09	-2.36941	-4.30235	-1.88399	-2.14014
1.84E+09	-2.45318	-4.5167	-1.88139	-2.15297
1.87E+09	-2.49659	-4.78832	-1.89955	-2.20286
1.9E+09	-2.58324	-5.03363	-1.88319	-2.28231
1.93E+09	-2.69544	-5.13222	-1.85967	-2.27735
1.96E+09	-2.73097	-5.29776	-1.80166	-2.3611
1.99E+09	-2.77982	-5.35541	-1.72981	-2.44817
2.01E+09	-2.91572	-5.35885	-1.68333	-2.44711
2.04E+09	-3.05131	-5.35757	-1.66244	-2.47734
2.07E+09	-3.28527	-5.34569	-1.5968	-2.5256
2.1E+09	-3.58801	-5.26396	-1.54984	-2.63012

2.13E+09	-3.97123	-5.1468	-1.53281	-2.72121
2.16E+09	-4.44554	-5.02848	-1.46141	-2.73385
2.19E+09	-4.9927	-4.86229	-1.46434	-2.89329
2.22E+09	-5.60738	-4.958	-1.44061	-3.0022
2.25E+09	-6.23479	-4.74212	-1.40781	-3.12373
2.28E+09	-7.04693	-4.82064	-1.39803	-3.34832
2.31E+09	-8.16016	-4.90497	-1.38198	-3.54441
2.34E+09	-9.45978	-4.91293	-1.35871	-3.76523
2.37E+09	-11.1999	-5.0594	-1.33814	-4.03022
2.4E+09	-13.0826	-5.01013	-1.29961	-4.14955
2.43E+09	-15.9004	-4.95588	-1.31053	-4.36842
2.46E+09	-19.6431	-5.03574	-1.32785	-4.50927
2.49E+09	-23.8283	-4.88742	-1.32802	-4.63188
2.52E+09	-21.4861	-5.00465	-1.34202	-4.92471
2.55E+09	-17.3236	-5.10916	-1.36139	-5.12496
2.58E+09	-14.6863	-5.29264	-1.36932	-5.41872
2.61E+09	-12.5246	-5.49381	-1.39687	-5.66362
2.64E+09	-10.7985	-5.74407	-1.44193	-5.9098
2.67E+09	-9.38215	-6.08841	-1.4845	-6.22818
2.7E+09	-8.34129	-6.54913	-1.53582	-6.44123
2.73E+09	-7.33161	-6.91608	-1.55772	-6.52533
2.76E+09	-6.58391	-7.50048	-1.5946	-6.63618
2.79E+09	-6.07777	-8.16666	-1.63979	-6.91411
2.82E+09	-5.62345	-8.96734	-1.65845	-7.13974
2.85E+09	-5.32504	-9.94794	-1.69178	-7.26675
2.88E+09	-5.01656	-11.2612	-1.76319	-7.62156
2.91E+09	-4.88053	-12.633	-1.80014	-7.7053
2.94E+09	-4.70478	-14.7882	-1.89347	-7.77313
2.97E+09	-4.59815	-18.1321	-1.96798	-7.96469
3E+09	-4.2695	-21.3791	-2.02638	-7.97262
3.03E+09	-4.04804	-27.0038	-2.10069	-8.05475
3.06E+09	-3.73149	-39.6992	-2.17946	-8.2806
3.09E+09	-3.51706	-27.7395	-2.22126	-8.5965
3.12E+09	-3.36536	-22.8803	-2.30795	-8.98529
3.15E+09	-3.19731	-21.0048	-2.37963	-9.28751
3.18E+09	-3.09112	-19.3084	-2.4742	-9.62135
3.21E+09	-2.98897	-17.5821	-2.57291	-9.88586
3.24E+09	-2.86716	-16.6539	-2.68325	-10.1541
3.27E+09	-2.78471	-15.9094	-2.74	-10.1293
3.3E+09	-2.70241	-14.9345	-2.84217	-10.1932
3.33E+09	-2.62426	-14.3322	-2.97855	-10.2811

3.36E+09	-2.63325	-14.0249	-3.06694	-10.3201
3.39E+09	-2.60846	-13.847	-3.19507	-10.5173
3.42E+09	-2.59885	-14.0337	-3.35419	-10.4582
3.45E+09	-2.5704	-13.5299	-3.48268	-10.4946
3.48E+09	-2.5417	-13.212	-3.59759	-10.8825
3.51E+09	-2.46137	-12.4735	-3.72899	-11.1247
3.54E+09	-2.39091	-11.458	-3.85551	-11.1402
3.57E+09	-2.33432	-10.5607	-4.01663	-10.9931
3.6E+09	-2.28448	-9.59186	-4.19215	-11.1346
3.63E+09	-2.25457	-8.87051	-4.38198	-11.4555
3.66E+09	-2.25119	-7.98604	-4.57102	-11.659
3.69E+09	-2.23327	-7.3081	-4.84142	-12.1207
3.72E+09	-2.26979	-6.62703	-5.12113	-12.6692
3.75E+09	-2.25764	-6.26818	-5.46388	-13.1382
3.78E+09	-2.25587	-5.84448	-5.7788	-14.2516
3.81E+09	-2.24823	-5.5738	-6.17429	-15.9009
3.84E+09	-2.23378	-5.4407	-6.61419	-17.9483
3.87E+09	-2.22917	-5.35887	-7.10344	-19.4582
3.9E+09	-2.22725	-5.37742	-7.66195	-17.5516
3.93E+09	-2.23834	-5.35935	-8.30143	-14.3753
3.96E+09	-2.20948	-5.39688	-9.14373	-11.3532
3.99E+09	-2.21783	-5.50281	-10.1013	-9.13542
4.01E+09	-2.23191	-5.61761	-11.2364	-7.7424
4.04E+09	-2.24208	-5.67154	-12.5114	-6.63412
4.07E+09	-2.26018	-5.70061	-14.0048	-5.82559
4.1E+09	-2.23166	-5.69324	-15.6448	-5.21616
4.13E+09	-2.26161	-5.68056	-17.4129	-4.7695
4.16E+09	-2.3165	-5.62719	-19.216	-4.43182
4.19E+09	-2.32294	-5.68176	-20.8005	-4.10822
4.22E+09	-2.40805	-5.76957	-22.186	-3.87592
4.25E+09	-2.43375	-5.75931	-22.1213	-3.66854
4.28E+09	-2.47306	-5.8036	-21.1695	-3.49975
4.31E+09	-2.60207	-5.93591	-19.5327	-3.34071
4.34E+09	-2.68725	-6.01805	-17.9366	-3.20507
4.37E+09	-2.78832	-6.13884	-16.6296	-3.06508
4.4E+09	-2.79405	-6.30841	-15.5788	-2.92872
4.43E+09	-2.55806	-6.57057	-14.8985	-2.76618
4.46E+09	-2.24139	-6.94513	-17.1269	-2.64856
4.49E+09	-2.00067	-6.82939	-26.2638	-2.47348
4.52E+09	-1.89545	-6.62882	-28.742	-2.28708
4.55E+09	-1.88838	-6.49754	-21.7761	-2.10748

4.58E+09	-1.91263	-6.39023	-18.4299	-1.94516
4.61E+09	-1.94001	-6.25247	-16.9995	-1.78951
4.64E+09	-1.98021	-6.1403	-15.9626	-1.66338
4.67E+09	-2.04453	-6.1007	-15.2544	-1.55536
4.7E+09	-2.09496	-5.95563	-14.7883	-1.46676
4.73E+09	-2.14865	-5.91129	-14.0846	-1.41603
4.76E+09	-2.21484	-5.83357	-13.4678	-1.38631
4.79E+09	-2.29847	-5.7347	-12.9435	-1.38633
4.82E+09	-2.37591	-5.60204	-12.4065	-1.44758
4.85E+09	-2.49314	-5.52783	-11.9794	-1.55437
4.88E+09	-2.58234	-5.44664	-11.4717	-1.68818
4.91E+09	-2.65808	-5.40696	-11.2639	-1.87384
4.94E+09	-2.76882	-5.40721	-11.037	-2.08604
4.97E+09	-2.83104	-5.34822	-10.9621	-2.34185
5E+09	-2.95033	-5.39149	-10.7154	-2.63726
5.03E+09	-3.03902	-5.36627	-10.4795	-2.89227
5.06E+09	-3.07129	-5.37122	-10.2239	-3.13003
5.09E+09	-3.13449	-5.35786	-9.93064	-3.37355
5.12E+09	-3.16847	-5.37829	-9.62078	-3.59137
5.15E+09	-3.21207	-5.38614	-9.48555	-3.8047
5.18E+09	-3.28135	-5.36653	-9.1994	-4.03762
5.21E+09	-3.27655	-5.42212	-9.08706	-4.33451
5.24E+09	-3.28413	-5.46752	-8.95253	-4.60785
5.27E+09	-3.29075	-5.50955	-8.91255	-4.90776
5.3E+09	-3.24177	-5.48403	-8.76486	-5.20962
5.33E+09	-3.19792	-5.47194	-8.62956	-5.48426
5.36E+09	-3.15239	-5.42357	-8.53273	-5.76706
5.39E+09	-3.06423	-5.35215	-8.40001	-6.07251
5.42E+09	-3.01546	-5.33471	-8.28513	-6.45691
5.45E+09	-2.96241	-5.30605	-8.13201	-6.84314
5.48E+09	-2.85392	-5.18592	-8.06149	-7.25209
5.51E+09	-2.79542	-5.13142	-7.99977	-7.68609
5.54E+09	-2.76333	-5.04748	-7.87566	-8.18958
5.57E+09	-2.76211	-4.98493	-7.79343	-8.72264
5.6E+09	-2.78912	-4.89363	-7.82409	-9.24792
5.63E+09	-2.8078	-4.82583	-7.7721	-9.88248
5.66E+09	-2.80288	-4.75197	-7.75741	-10.5529
5.69E+09	-2.79523	-4.73293	-7.68737	-11.3072
5.72E+09	-2.80945	-4.68443	-7.68332	-12.1403
5.75E+09	-2.85832	-4.67826	-7.70226	-12.7955
5.78E+09	-2.88625	-4.64986	-7.6658	-13.8027

5.81E+09	-2.91917	-4.58785	-7.61011	-14.9103
5.84E+09	-2.99624	-4.58862	-7.56185	-15.7693
5.87E+09	-3.06386	-4.57151	-7.61596	-16.7115
5.9E+09	-3.09654	-4.54646	-7.66098	-17.6217
5.93E+09	-3.18347	-4.53119	-7.67126	-18.5365
5.96E+09	-3.23616	-4.45618	-7.66218	-19.3042
5.99E+09	-3.34289	-4.38647	-7.72226	-19.9155
6.01E+09	-3.43274	-4.31136	-7.78021	-20.7232
6.04E+09	-3.60268	-4.23425	-7.82483	-21.2133
6.07E+09	-3.81945	-4.17758	-7.92989	-21.2913
6.1E+09	-3.9293	-4.10759	-8.08477	-21.1195
6.13E+09	-4.15617	-4.05622	-8.16109	-20.6847
6.16E+09	-4.31587	-3.98454	-8.23855	-20.9255
6.19E+09	-4.40851	-3.9064	-8.23525	-20.1091
6.22E+09	-4.62074	-3.84549	-8.32828	-19.7267
6.25E+09	-4.79583	-3.81831	-8.53596	-19.2354
6.28E+09	-5.0365	-3.78754	-8.65871	-18.6275
6.31E+09	-5.26426	-3.80762	-8.88919	-18.3578
6.34E+09	-5.44927	-3.77312	-8.96053	-17.8105
6.37E+09	-5.67462	-3.77289	-9.14752	-17.2144
6.4E+09	-6.02736	-3.80281	-9.32727	-16.8483
6.43E+09	-6.41831	-3.83247	-9.48989	-16.1275
6.46E+09	-6.7613	-3.9077	-9.76143	-15.5202
6.49E+09	-7.07264	-4.0366	-10.0315	-15.2236
6.52E+09	-7.31976	-4.11903	-10.1935	-14.7578
6.55E+09	-7.62579	-4.19475	-10.397	-14.4888
6.58E+09	-8.03131	-4.27575	-10.5001	-14.021
6.61E+09	-8.42602	-4.34924	-10.7469	-13.6611
6.64E+09	-8.88607	-4.47324	-11.0374	-13.3376
6.67E+09	-9.36451	-4.62222	-11.2721	-13.3066
6.7E+09	-9.7843	-4.7751	-11.5375	-13.0602
6.73E+09	-10.1742	-4.93411	-11.7615	-13.0026
6.76E+09	-10.7224	-5.08777	-11.9578	-13.0976
6.79E+09	-11.3726	-5.2551	-12.2068	-12.693
6.82E+09	-12.0855	-5.45175	-12.4856	-12.7934
6.85E+09	-12.7519	-5.70033	-12.9174	-12.9286
6.88E+09	-13.2195	-5.91747	-13.1861	-12.847
6.91E+09	-13.6648	-6.06465	-13.3619	-13.2993
6.94E+09	-13.9168	-6.24068	-13.3986	-13.2665

### B3. Planar Ultra-Wideband Antenna

freq	0100	1000	1111	0000	0111	1011	0011	1100
1.00E+09	-1.11	-2.41	-5.31	-5.23	-2.51	-5.80	-3.24	-3.64
1.05E+09	-1.11	-2.44	-5.21	-5.26	-2.52	-6.00	-3.10	-3.72
1.10E+09	-1.11	-2.50	-5.12	-5.29	-2.58	-6.10	-2.96	-3.83
1.15E+09	-1.11	-2.51	-5.12	-5.28	-2.63	-6.35	-2.96	-3.95
1.20E+09	-1.13	-2.57	-5.19	-5.27	-2.79	-6.61	-3.05	-4.08
1.25E+09	-1.18	-2.63	-5.25	-5.30	-2.89	-6.91	-3.20	-4.21
1.30E+09	-1.25	-2.66	-5.25	-5.35	-2.89	-7.28	-3.38	-4.27
1.35E+09	-1.36	-2.68	-5.36	-5.49	-3.00	-7.60	-3.66	-4.37
1.40E+09	-1.47	-2.72	-5.54	-5.64	-3.19	-7.86	-3.97	-4.46
1.45E+09	-1.58	-2.84	-5.72	-5.76	-3.47	-8.21	-4.15	-4.54
1.50E+09	-1.71	-3.01	-5.78	-5.79	-3.79	-8.66	-4.22	-4.57
1.55E+09	-1.93	-3.29	-5.83	-5.78	-4.14	-8.91	-4.33	-4.62
1.60E+09	-2.25	-3.47	-6.06	-5.82	-4.51	-8.97	-4.67	-4.76
1.65E+09	-2.65	-4.19	-6.50	-5.94	-4.90	-9.03	-5.22	-4.95
1.70E+09	-3.10	-4.70	-7.00	-6.20	-5.35	-9.08	-5.92	-5.18
1.75E+09	-3.57	-5.02	-7.49	-6.52	-5.80	-9.15	-6.83	-5.44
1.80E+09	-4.20	-5.60	-8.17	-6.89	-6.18	-9.25	-7.96	-5.80
1.85E+09	-5.02	-6.71	-8.78	-7.26	-6.53	-9.31	-9.25	-6.17
1.90E+09	-6.01	-7.26	-9.43	-7.76	-6.88	-9.39	-10.55	-6.57
1.95E+09	-6.90	-7.55	-10.01	-8.48	-7.33	-9.46	-11.96	-6.98
2.00E+09	-7.81	-8.11	-10.79	-9.51	-7.95	-9.79	-13.49	-7.47
2.05E+09	-8.70	-8.87	-11.61	-11.03	-8.85	-10.47	-15.19	-7.99
2.10E+09	-9.84	-9.42	-12.37	-12.79	-10.13	-11.02	-17.00	-8.51
2.15E+09	-10.98	-10.30	-13.07	-14.38	-11.54	-12.00	-18.98	-9.02
2.20E+09	-11.92	-11.78	-13.60	-15.31	-13.28	-13.25	-21.28	-9.55
2.25E+09	-12.59	-13.96	-14.01	-15.83	-15.77	-14.74	-24.55	-10.08
2.30E+09	-13.02	-16.11	-14.30	-16.48	-20.37	-16.27	-27.07	-10.75
2.35E+09	-13.26	-17.72	-14.60	-18.20	-23.28	-17.63	-26.54	-11.42
2.40E+09	-12.90	-19.22	-14.64	-18.57	-21.64	-18.54	-21.89	-12.08
2.45E+09	-11.15	-19.64	-14.57	-15.07	-15.47	-18.87	-14.44	-12.82
2.50E+09	-7.58	-19.11	-14.34	-9.18	-8.27	-18.68	-8.25	-13.68
2.55E+09	-4.69	-17.61	-14.22	-4.61	-4.38	-17.72	-4.69	-14.36
2.60E+09	-3.94	-15.59	-14.06	-6.05	-6.70	-16.62	-7.18	-14.62
2.65E+09	-5.62	-14.01	-13.92	-8.14	-8.99	-15.14	-9.62	-14.47
2.70E+09	-7.30	-12.46	-13.77	-9.26	-9.58	-14.19	-10.58	-14.11
2.75E+09	-8.43	-11.60	-13.70	-9.91	-11.15	-13.16	-10.80	-13.48
2.80E+09	-9.54	-11.56	-13.60	-10.37	-11.82	-12.36	-10.60	-12.75



2.85E+09	-10.70	-11.83	-13.41	-10.78	-11.91	-11.73	-10.53	-12.00
2.90E+09	-11.54	-11.86	-13.20	-11.29	-11.71	-11.43	-10.55	-11.51
2.95E+09	-11.80	-11.48	-12.94	-11.82	-11.58	-11.41	-10.71	-11.06
3.00E+09	-11.85	-11.47	-12.80	-12.54	-11.62	-11.48	-10.97	-10.77
3.05E+09	-12.00	-11.49	-12.60	-13.12	-11.90	-11.82	-11.54	-10.50
3.10E+09	-12.26	-12.05	-12.49	-13.50	-12.19	-12.15	-12.30	-10.37
3.15E+09	-12.25	-11.88	-12.35	-13.38	-12.24	-12.55	-13.04	-10.24
3.20E+09	-12.03	-12.31	-12.21	-13.00	-12.10	-12.79	-13.67	-10.27
3.25E+09	-11.68	-12.02	-11.95	-12.56	-11.73	-12.74	-13.98	-10.27
3.30E+09	-11.52	-11.89	-11.68	-12.07	-11.56	-12.48	-13.90	-10.43
3.35E+09	-11.54	-11.61	-11.39	-11.59	-11.50	-11.71	-13.25	-10.53
3.40E+09	-11.64	-11.16	-11.24	-10.89	-11.51	-10.79	-12.20	-10.80
3.45E+09	-11.71	-10.25	-11.08	-10.07	-11.31	-9.52	-10.86	-11.12
3.50E+09	-11.65	-8.42	-10.94	-8.82	-10.75	-7.88	-9.15	-11.52
3.55E+09	-11.67	-5.97	-10.78	-6.82	-10.41	-5.75	-6.74	-11.91
3.60E+09	-11.81	-4.30	-10.72	-4.39	-10.32	-4.53	-4.83	-12.20
3.65E+09	-12.12	-5.17	-10.72	-5.83	-10.23	-5.37	-5.43	-12.44
3.70E+09	-12.36	-7.29	-10.80	-7.80	-10.36	-7.53	-7.59	-12.57
3.75E+09	-12.62	-9.40	-10.82	-9.78	-10.49	-9.43	-9.81	-12.60
3.80E+09	-12.85	-10.54	-10.91	-10.60	-10.53	-10.27	-10.78	-12.51
3.85E+09	-13.19	-11.09	-10.97	-11.15	-10.44	-10.66	-11.10	-12.43
3.90E+09	-13.37	-11.62	-11.06	-11.56	-10.57	-10.90	-11.15	-12.43
3.95E+09	-13.50	-11.88	-11.08	-12.05	-10.77	-11.17	-11.08	-12.52
4.00E+09	-13.51	-12.40	-11.10	-12.41	-11.10	-11.43	-11.05	-12.61
4.05E+09	-13.67	-12.49	-11.18	-12.87	-11.36	-11.52	-11.00	-12.73
4.10E+09	-13.89	-12.43	-11.18	-13.17	-11.58	-11.53	-10.96	-12.99
4.15E+09	-14.28	-12.45	-11.21	-13.42	-11.89	-11.46	-10.95	-13.31
4.20E+09	-14.60	-12.28	-11.24	-13.64	-12.21	-11.53	-10.97	-13.79
4.25E+09	-14.82	-12.57	-11.44	-13.94	-12.72	-11.61	-10.97	-14.18
4.30E+09	-15.04	-12.58	-11.54	-14.39	-13.03	-11.61	-11.00	-14.70
4.35E+09	-15.49	-12.98	-11.72	-14.64	-13.44	-11.66	-11.13	-15.19
4.40E+09	-16.26	-13.44	-11.89	-14.90	-13.92	-11.83	-11.36	-15.89
4.45E+09	-17.07	-14.21	-12.21	-15.05	-14.70	-12.20	-11.61	-16.55
4.50E+09	-17.73	-14.88	-12.51	-15.38	-15.57	-12.45	-11.92	-17.23
4.55E+09	-18.42	-15.68	-12.83	-15.82	-16.03	-12.81	-12.27	-17.54
4.60E+09	-18.96	-15.89	-13.28	-16.40	-16.19	-13.19	-12.71	-17.77
4.65E+09	-19.24	-16.23	-13.72	-17.13	-16.17	-13.62	-13.19	-17.90
4.70E+09	-19.07	-16.38	-14.25	-17.71	-16.72	-14.03	-13.66	-18.12
4.75E+09	-18.75	-16.97	-14.75	-18.19	-17.59	-14.30	-14.26	-18.23
4.80E+09	-18.74	-17.31	-15.37	-18.57	-18.59	-14.77	-14.86	-18.29
4.85E+09	-18.66	-17.85	-16.03	-18.71	-18.88	-15.11	-15.48	-18.43

4.90E+09	-18.38	-17.94	-16.88	-18.92	-18.88	-15.55	-16.07	-18.63
4.95E+09	-17.59	-18.16	-17.77	-19.17	-18.73	-16.08	-16.73	-18.84
5.00E+09	-16.70	-18.19	-18.66	-19.62	-18.85	-16.72	-17.47	-18.81
5.05E+09	-16.01	-18.21	-19.22	-19.71	-19.03	-17.45	-18.27	-19.03
5.10E+09	-15.73	-18.24	-19.85	-19.86	-19.23	-18.27	-19.13	-19.13
5.15E+09	-15.68	-18.19	-20.55	-19.82	-19.43	-19.31	-20.04	-19.23
5.20E+09	-15.50	-18.14	-21.23	-19.98	-19.51	-20.69	-20.99	-18.90
5.25E+09	-15.06	-18.01	-21.73	-19.61	-19.33	-22.18	-22.15	-18.52
5.30E+09	-14.50	-17.55	-22.22	-18.90	-18.94	-23.98	-23.42	-17.91
5.35E+09	-14.02	-17.10	-23.21	-18.00	-18.44	-25.98	-24.67	-17.13
5.40E+09	-13.67	-16.56	-25.81	-17.07	-18.04	-28.30	-27.00	-16.35
5.45E+09	-13.41	-15.92	-25.67	-15.89	-17.98	-26.10	-22.80	-15.76
5.50E+09	-13.17	-15.53	-21.13	-14.49	-16.61	-20.36	-16.00	-15.22
5.55E+09	-12.92	-15.18	-4.50	-13.80	-15.06	-10.13	-11.79	-14.52
5.60E+09	-12.60	-15.01	-7.15	-13.64	-4.54	-4.51	-9.01	-13.87
5.65E+09	-12.25	-14.60	-9.20	-13.72	-6.31	-5.96	-4.44	-13.42
5.70E+09	-11.89	-14.29	-10.08	-13.30	-7.90	-7.29	-7.45	-13.11
5.75E+09	-11.72	-13.82	-11.95	-12.77	-9.88	-9.28	-9.03	-12.76
5.80E+09	-11.76	-13.41	-12.71	-12.40	-11.71	-12.56	-11.02	-12.41
5.85E+09	-11.91	-12.91	-13.52	-11.99	-12.03	-13.23	-12.67	-12.18
5.90E+09	-12.05	-12.56	-14.27	-11.76	-12.44	-13.83	-13.33	-12.07
5.95E+09	-12.10	-12.49	-15.96	-11.54	-12.93	-14.36	-14.94	-11.95
6.00E+09	-12.21	-12.53	-16.74	-11.48	-13.69	-15.03	-15.30	-11.85
6.05E+09	-12.34	-12.77	-17.46	-11.60	-14.04	-15.88	-16.54	-11.83
6.10E+09	-12.65	-12.90	-18.81	-11.96	-15.25	-16.76	-17.84	-11.94
6.15E+09	-13.07	-12.98	-20.51	-12.41	-18.38	-19.06	-18.30	-12.17
6.20E+09	-13.64	-13.14	-21.79	-12.91	-19.80	-18.38	-19.51	-12.50
6.25E+09	-14.30	-13.52	-17.43	-13.26	-19.28	-17.06	-20.38	-13.01
6.30E+09	-15.01	-14.01	-16.68	-13.60	-19.13	-16.71	-18.74	-13.59
6.35E+09	-15.75	-14.50	-16.53	-14.01	-19.21	-16.70	-18.23	-14.23
6.40E+09	-16.89	-15.14	-16.51	-14.79	-19.27	-16.79	-18.14	-15.05
6.45E+09	-18.40	-15.75	-16.41	-15.88	-19.39	-16.78	-18.17	-15.83
6.50E+09	-20.00	-16.35	-16.12	-17.08	-19.25	-16.78	-18.06	-16.44
6.55E+09	-20.89	-16.55	-15.52	-17.94	-18.55	-16.26	-17.64	-16.57
6.60E+09	-21.25	-16.67	-14.82	-18.23	-17.48	-15.60	-16.99	-16.51
6.65E+09	-21.04	-16.37	-14.21	-17.93	-16.40	-14.82	-16.28	-16.25
6.70E+09	-20.29	-16.06	-13.67	-17.09	-15.43	-14.12	-15.44	-15.64
6.75E+09	-18.82	-15.53	-13.07	-16.10	-14.45	-13.37	-14.58	-14.85
6.80E+09	-17.23	-15.07	-12.48	-15.16	-13.60	-12.63	-13.72	-14.02
6.85E+09	-15.98	-14.43	-11.96	-14.50	-13.37	-12.06	-13.03	-13.43
6.90E+09	-14.99	-13.72	-11.51	-13.85	-12.97	-11.49	-12.33	-12.78

6.95E+09	-14.17	-13.03	-11.04	-13.15	-12.79	-11.22	-11.72	-12.17
7.00E+09	-13.48	-12.56	-10.56	-12.37	-12.61	-10.96	-11.16	-11.62
7.05E+09	-12.94	-12.19	-10.15	-11.64	-12.46	-10.74	-10.73	-11.45
7.10E+09	-12.44	-11.89	-9.91	-11.13	-12.22	-10.54	-10.34	-11.30
7.15E+09	-12.02	-11.52	-9.83	-10.76	-12.04	-10.30	-10.01	-11.01
7.20E+09	-11.75	-11.23	-9.78	-10.54	-11.89	-10.19	-9.76	-10.81
7.25E+09	-11.55	-10.99	-9.61	-10.32	-11.56	-9.98	-9.71	-10.60
7.30E+09	-11.31	-10.84	-9.59	-10.17	-11.40	-9.97	-9.65	-10.41
7.35E+09	-11.06	-10.75	-9.53	-10.08	-11.11	-9.94	-9.63	-10.20
7.40E+09	-10.82	-10.66	-9.44	-9.92	-10.74	-9.91	-9.54	-10.10
7.45E+09	-10.68	-10.63	-9.40	-9.85	-10.50	-9.89	-9.37	-9.97
7.50E+09	-10.52	-10.58	-9.39	-9.74	-10.11	-9.87	-9.29	-10.10
7.55E+09	-10.40	-10.57	-9.40	-9.68	-9.82	-9.85	-9.25	-10.28
7.60E+09	-10.29	-10.53	-9.45	-9.66	-9.70	-9.82	-9.19	-10.47
7.65E+09	-10.22	-10.46	-9.49	-9.60	-9.62	-9.78	-9.18	-10.65
7.70E+09	-10.21	-10.41	-9.50	-9.59	-9.57	-9.77	-9.17	-10.85
7.75E+09	-10.25	-10.44	-9.52	-9.53	-9.55	-9.73	-9.19	-11.09
7.80E+09	-10.39	-10.52	-9.53	-9.49	-9.53	-9.77	-9.20	-11.28
7.85E+09	-10.54	-10.63	-9.56	-9.60	-9.60	-9.80	-9.21	-11.48
7.90E+09	-10.66	-10.74	-9.60	-9.61	-9.64	-9.84	-9.23	-11.61
7.95E+09	-10.77	-10.85	-9.65	-9.72	-9.68	-9.87	-9.36	-11.88
8.00E+09	-10.86	-10.99	-9.72	-9.99	-9.75	-9.92	-9.55	-12.09
8.05E+09	-10.91	-11.13	-9.80	-10.36	-9.85	-9.95	-9.78	-12.27
8.10E+09	-10.88	-11.36	-9.88	-10.78	-9.93	-9.98	-9.91	-12.50
8.15E+09	-10.89	-11.56	-9.99	-11.26	-10.03	-10.13	-10.75	-12.66
8.20E+09	-10.93	-11.85	-10.03	-11.69	-10.39	-10.40	-10.82	-12.89
8.25E+09	-11.00	-12.15	-10.40	-12.07	-10.73	-10.65	-10.95	-13.00
8.30E+09	-11.08	-12.49	-10.74	-12.39	-11.06	-10.96	-11.25	-13.22
8.35E+09	-11.21	-12.78	-10.98	-12.73	-11.41	-11.24	-11.51	-13.46
8.40E+09	-11.46	-13.10	-11.10	-13.12	-11.78	-11.50	-11.62	-13.67
8.45E+09	-11.75	-13.46	-11.24	-13.55	-12.08	-11.85	-11.78	-13.81
8.50E+09	-12.18	-13.94	-11.30	-14.01	-12.20	-12.12	-11.89	-14.03
8.55E+09	-12.69	-14.56	-11.35	-14.52	-12.24	-12.34	-11.96	-14.23
8.60E+09	-13.32	-15.12	-11.40	-15.10	-12.30	-12.50	-12.07	-14.40
8.65E+09	-13.96	-15.59	-11.45	-15.68	-12.43	-12.69	-12.14	-14.57
8.70E+09	-14.58	-16.02	-11.79	-16.25	-12.65	-12.92	-12.21	-15.11
8.75E+09	-15.32	-16.64	-12.19	-16.84	-12.92	-13.19	-12.65	-15.63
8.80E+09	-16.19	-17.45	-12.64	-17.57	-13.28	-13.30	-13.14	-16.38
8.85E+09	-17.26	-18.36	-13.22	-18.34	-13.71	-13.56	-13.70	-17.18
8.90E+09	-18.54	-19.48	-13.89	-19.56	-14.24	-14.06	-14.32	-18.50
8.95E+09	-20.19	-21.22	-14.41	-21.53	-14.98	-14.64	-15.03	-20.01

9.00E+09	-22.50	-22.89	-15.15	-24.37	-15.79	-15.35	-15.81	-21.82
9.05E+09	-25.90	-24.75	-15.85	-27.42	-16.65	-16.09	-16.60	-23.88
9.10E+09	-33.06	-26.22	-16.83	-30.07	-17.48	-16.86	-17.61	-25.81
9.15E+09	-35.73	-26.89	-18.79	-30.05	-18.54	-17.81	-18.75	-26.68
9.20E+09	-34.67	-25.76	-20.21	-28.15	-19.72	-18.94	-20.06	-25.73
9.25E+09	-27.70	-23.66	-21.94	-24.60	-21.20	-20.13	-21.37	-23.61
9.30E+09	-23.28	-21.42	-24.15	-21.72	-22.81	-21.62	-23.02	-21.27
9.35E+09	-20.28	-19.61	-26.71	-19.33	-24.40	-23.26	-24.58	-19.22
9.40E+09	-18.07	-17.65	-28.65	-17.51	-25.40	-24.75	-25.90	-17.54
9.45E+09	-16.27	-15.95	-28.86	-16.11	-25.41	-25.58	-26.39	-16.05
9.50E+09	-14.66	-14.49	-27.12	-14.82	-24.43	-25.02	-25.77	-14.73
9.55E+09	-13.35	-13.26	-24.40	-13.68	-22.84	-23.77	-24.33	-13.49
9.60E+09	-12.20	-12.25	-21.75	-12.63	-21.14	-21.99	-22.34	-12.42
9.65E+09	-11.28	-11.41	-19.58	-11.68	-19.64	-20.61	-20.65	-11.53
9.70E+09	-10.44	-10.70	-17.78	-10.80	-18.30	-19.28	-19.05	-10.78
9.75E+09	-9.75	-10.01	-16.35	-10.08	-17.10	-17.98	-17.72	-10.08
9.80E+09	-9.15	-9.34	-15.22	-9.49	-16.01	-16.73	-16.56	-9.43
9.85E+09	-8.65	-8.76	-14.30	-8.99	-15.07	-15.62	-15.59	-8.86
9.90E+09	-8.19	-8.27	-13.51	-8.57	-14.32	-14.81	-14.77	-8.41
9.95E+09	-7.78	-7.90	-12.90	-8.17	-13.71	-14.19	-14.11	-8.03
1.00E+10	-7.43	-7.61	-12.44	-7.83	-13.28	-13.73	-13.58	-7.71
1.01E+10	-7.16	-7.36	-12.09	-7.50	-12.94	-13.29	-13.15	-7.41
1.01E+10	-6.92	-7.09	-11.78	-7.20	-12.61	-12.80	-12.78	-7.16
1.02E+10	-6.69	-6.85	-11.46	-6.92	-12.22	-12.37	-12.43	-6.90
1.02E+10	-6.44	-6.58	-11.23	-6.69	-11.86	-12.02	-12.12	-6.66
1.03E+10	-6.22	-6.39	-11.09	-6.51	-11.57	-11.89	-11.93	-6.42
1.03E+10	-6.03	-6.18	-11.00	-6.32	-11.38	-11.74	-11.76	-6.19
1.04E+10	-5.82	-5.97	-10.85	-6.10	-11.18	-11.51	-11.58	-5.96
1.04E+10	-5.61	-5.77	-10.71	-5.85	-11.06	-11.28	-11.41	-5.74
1.05E+10	-5.38	-5.57	-10.64	-5.59	-11.04	-11.20	-11.28	-5.52
1.05E+10	-5.16	-5.41	-10.53	-5.33	-11.00	-11.13	-11.11	-5.31
1.06E+10	-5.00	-5.35	-10.37	-5.21	-10.98	-10.99	-10.89	-5.17
1.06E+10	-5.05	-5.55	-10.23	-5.39	-11.03	-10.83	-10.72	-5.26
1.07E+10	-5.50	-6.22	-10.13	-6.08	-11.12	-10.68	-10.62	-5.78
1.07E+10	-6.70	-7.71	-9.97	-7.58	-11.11	-10.49	-10.46	-7.15
1.08E+10	-9.05	-10.41	-9.75	-10.25	-10.98	-10.28	-10.26	-9.82
1.08E+10	-12.47	-14.00	-9.56	-13.79	-10.88	-10.16	-10.10	-13.63
1.09E+10	-15.45	-16.27	-9.40	-16.29	-10.70	-10.01	-9.98	-16.29
1.09E+10	-16.37	-16.39	-9.18	-16.52	-10.43	-9.75	-9.79	-16.56
1.10E+10	-15.16	-14.52	-8.93	-14.86	-10.10	-9.45	-9.59	-14.72

#### B4. Reconfigurable filter

freq	SW1	SW2	SW3	SW4	SW5	All off
4.00E+09	-0.60	-0.17	-0.59	-0.59	-0.60	-0.68
4.12E+09	-0.63	-0.17	-0.62	-0.62	-0.63	-0.67
4.23E+09	-0.65	-0.18	-0.65	-0.65	-0.66	-0.66
4.35E+09	-0.68	-0.18	-0.68	-0.68	-0.69	-0.66
4.46E+09	-0.71	-0.19	-0.71	-0.71	-0.72	-0.66
4.58E+09	-0.74	-0.19	-0.74	-0.75	-0.75	-0.66
4.69E+09	-0.77	-0.20	-0.77	-0.78	-0.78	-0.66
4.81E+09	-0.80	-0.20	-0.80	-0.82	-0.81	-0.66
4.92E+09	-0.83	-0.20	-0.83	-0.85	-0.84	-0.66
5.04E+09	-0.86	-0.21	-0.86	-0.89	-0.87	-0.67
5.15E+09	-0.90	-0.21	-0.90	-0.92	-0.91	-0.67
5.27E+09	-0.93	-0.21	-0.93	-0.95	-0.94	-0.68
5.38E+09	-0.96	-0.22	-0.96	-0.99	-0.97	-0.70
5.50E+09	-0.99	-0.22	-0.99	-1.02	-1.00	-0.71
5.62E+09	-1.02	-0.22	-1.02	-1.05	-1.03	-0.73
5.73E+09	-1.06	-0.23	-1.06	-1.08	-1.06	-0.75
5.85E+09	-1.09	-0.23	-1.09	-1.11	-1.09	-0.77
5.96E+09	-1.12	-0.23	-1.12	-1.14	-1.12	-0.79
6.08E+09	-1.15	-0.24	-1.15	-1.17	-1.15	-0.82
6.19E+09	-1.19	-0.24	-1.18	-1.20	-1.18	-0.85
6.31E+09	-1.22	-0.24	-1.21	-1.23	-1.21	-0.89
6.42E+09	-1.26	-0.24	-1.24	-1.26	-1.24	-0.93
6.54E+09	-1.29	-0.24	-1.28	-1.29	-1.26	-0.97
6.65E+09	-1.33	-0.25	-1.31	-1.32	-1.29	-1.01
6.77E+09	-1.36	-0.25	-1.34	-1.34	-1.32	-1.06
6.88E+09	-1.40	-0.25	-1.37	-1.37	-1.35	-1.12
7.00E+09	-1.44	-0.25	-1.40	-1.40	-1.37	-1.18
7.12E+09	-1.48	-0.26	-1.44	-1.43	-1.40	-1.25
7.23E+09	-1.53	-0.26	-1.47	-1.46	-1.43	-1.32
7.35E+09	-1.58	-0.26	-1.51	-1.49	-1.45	-1.40
7.46E+09	-1.63	-0.26	-1.55	-1.53	-1.48	-1.49
7.58E+09	-1.69	-0.27	-1.59	-1.56	-1.51	-1.60
7.69E+09	-1.76	-0.27	-1.63	-1.60	-1.54	-1.71
7.81E+09	-1.83	-0.27	-1.68	-1.64	-1.58	-1.85
7.92E+09	-1.91	-0.28	-1.74	-1.68	-1.61	-2.00
8.04E+09	-2.00	-0.28	-1.79	-1.73	-1.65	-2.18
8.15E+09	-2.11	-0.28	-1.86	-1.78	-1.69	-2.40
8.27E+09	-2.24	-0.29	-1.93	-1.84	-1.73	-2.64
8.38E+09	-2.39	-0.29	-2.01	-1.90	-1.78	-2.98

8.50E+09	-2.57	-0.30	-2.10	-1.97	-1.83	-3.33
8.62E+09	-2.79	-0.30	-2.21	-2.05	-1.89	-3.87
8.73E+09	-3.05	-0.31	-2.32	-2.14	-1.96	-4.46
8.85E+09	-3.37	-0.32	-2.46	-2.23	-2.03	-5.38
8.96E+09	-3.77	-0.33	-2.61	-2.34	-2.10	-6.47
9.08E+09	-4.29	-0.34	-2.79	-2.47	-2.19	-8.14
9.19E+09	-4.95	-0.35	-2.99	-2.60	-2.28	-10.49
9.31E+09	-5.82	-0.36	-3.24	-2.76	-2.39	-12.98
9.42E+09	-6.99	-0.37	-3.52	-2.94	-2.51	-15.48
9.54E+09	-8.61	-0.39	-3.85	-3.15	-2.63	-15.53
9.65E+09	-10.91	-0.41	-4.24	-3.38	-2.78	-11.14
9.77E+09	-14.19	-0.44	-4.71	-3.65	-2.94	-8.28
9.88E+09	-17.51	-0.47	-5.28	-3.95	-3.12	-6.32
1.00E+10	-15.74	-0.51	-5.97	-4.31	-3.31	-4.72
1.01E+10	-11.81	-0.55	-6.82	-4.71	-3.53	-3.86
1.02E+10	-8.86	-0.61	-7.86	-5.19	-3.78	-3.07
1.03E+10	-6.79	-0.69	-9.17	-5.75	-4.05	-2.61
1.05E+10	-5.34	-0.79	-10.80	-6.40	-4.36	-2.22
1.06E+10	-4.29	-0.93	-12.79	-7.16	-4.70	-1.96
1.07E+10	-3.53	-1.13	-14.96	-8.06	-5.09	-1.77
1.08E+10	-2.98	-1.44	-16.29	-9.13	-5.52	-1.62
1.09E+10	-2.57	-1.93	-15.45	-10.38	-6.01	-1.52
1.10E+10	-2.26	-2.80	-13.27	-11.83	-6.56	-1.44
1.12E+10	-2.03	-4.57	-11.10	-13.41	-7.18	-1.40
1.13E+10	-1.86	-8.99	-9.28	-14.85	-7.87	-1.36
1.14E+10	-1.73	-38.28	-7.83	-15.58	-8.64	-1.35
1.15E+10	-1.64	-7.77	-6.68	-15.12	-9.51	-1.33
1.16E+10	-1.57	-3.32	-5.75	-13.79	-10.46	-1.34
1.17E+10	-1.52	-1.65	-5.01	-12.21	-11.49	-1.35
1.18E+10	-1.48	-0.91	-4.42	-10.73	-12.54	-1.36
1.20E+10	-1.46	-0.54	-3.94	-9.44	-13.54	-1.38
1.21E+10	-1.45	-0.34	-3.55	-8.34	-14.32	-1.41
1.22E+10	-1.45	-0.22	-3.23	-7.41	-14.69	-1.44
1.23E+10	-1.45	-0.15	-2.97	-6.64	-14.54	-1.47
1.24E+10	-1.47	-0.11	-2.76	-5.98	-13.95	-1.51
1.25E+10	-1.48	-0.07	-2.59	-5.43	-13.09	-1.54
1.27E+10	-1.50	-0.05	-2.44	-4.96	-12.12	-1.58
1.28E+10	-1.52	-0.04	-2.32	-4.56	-11.17	-1.62
1.29E+10	-1.55	-0.03	-2.23	-4.22	-10.28	-1.67
1.30E+10	-1.58	-0.02	-2.15	-3.93	-9.46	-1.71
1.31E+10	-1.61	-0.02	-2.08	-3.68	-8.73	-1.76

1.32E+10	-1.64	-0.01	-2.03	-3.46	-8.07	-1.81
1.33E+10	-1.67	-0.01	-1.99	-3.27	-7.48	-1.86
1.35E+10	-1.71	-0.01	-1.95	-3.11	-6.96	-1.91
1.36E+10	-1.74	0.00	-1.92	-2.97	-6.49	-1.96
1.37E+10	-1.78	0.00	-1.90	-2.84	-6.07	-2.01
1.38E+10	-1.82	0.00	-1.88	-2.73	-5.70	-2.07
1.39E+10	-1.86	0.00	-1.87	-2.64	-5.36	-2.13
1.40E+10	-1.90	0.00	-1.86	-2.55	-5.06	-2.20
1.42E+10	-1.94	0.00	-1.85	-2.48	-4.78	-2.26
1.43E+10	-1.98	0.00	-1.84	-2.41	-4.53	-2.33
1.44E+10	-2.03	0.00	-1.84	-2.35	-4.30	-2.41
1.45E+10	-2.08	0.00	-1.84	-2.30	-4.10	-2.48
1.46E+10	-2.12	0.00	-1.84	-2.25	-3.91	-2.57
1.47E+10	-2.17	0.00	-1.84	-2.21	-3.74	-2.65
1.48E+10	-2.22	0.00	-1.85	-2.17	-3.58	-2.74
1.50E+10	-2.28	0.00	-1.85	-2.14	-3.44	-2.83
1.51E+10	-2.33	0.00	-1.86	-2.11	-3.31	-2.92
1.52E+10	-2.39	0.00	-1.86	-2.09	-3.19	-3.01
1.53E+10	-2.44	0.00	-1.87	-2.07	-3.08	-3.11
1.54E+10	-2.50	0.00	-1.88	-2.05	-2.98	-3.20
1.55E+10	-2.56	0.00	-1.89	-2.03	-2.89	-3.30
1.57E+10	-2.61	0.00	-1.90	-2.02	-2.81	-3.40
1.58E+10	-2.67	0.00	-1.91	-2.00	-2.73	-3.50
1.59E+10	-2.72	0.00	-1.92	-1.99	-2.67	-3.61
1.60E+10	-2.78	0.00	-1.93	-1.97	-2.60	-3.71

## B5. Star Antenna

Freq [GHz]	S1S2S3	S1S2S3S4	S1S2S3S4S5	S1S2S3S4S6	S1S2S3S5S6
0.80000	-0.11589	-0.10423	-0.08703	-0.10701	-0.11327
0.82100	-0.12597	-0.11100	-0.08945	-0.11443	-0.12268
0.84200	-0.13940	-0.11989	-0.09243	-0.12391	-0.13503
0.86300	-0.15764	-0.13170	-0.09607	-0.13618	-0.15147
0.88400	-0.18303	-0.14775	-0.10049	-0.15231	-0.17379
0.90500	-0.21963	-0.17021	-0.10587	-0.17402	-0.20492
0.92600	-0.27491	-0.20292	-0.11241	-0.20416	-0.24981
0.94700	-0.36398	-0.25327	-0.12040	-0.24783	-0.31751
0.96800	-0.52162	-0.33685	-0.13025	-0.31492	-0.42574
0.98900	-0.84433	-0.49158	-0.14251	-0.42685	-0.61287
1.01000	-1.69428	-0.82949	-0.15800	-0.63775	-0.97272
1.03100	-5.39054	-1.79658	-0.17807	-1.11702	-1.77228
1.05200	-12.14913	-6.26652	-0.20545	-2.59758	-3.86682
1.07300	-7.86146	-11.38399	-0.24769	-10.47700	-8.19805
1.09400	-12.89525	-5.59222	-0.27470	-7.08453	-5.13917
1.11500	-3.23033	-2.73805	-0.33345	-2.48870	-2.29602
1.13600	-1.45852	-0.95792	-0.42301	-1.59126	-1.23672
1.15700	-0.94277	-0.58006	-0.56025	-1.50438	-0.78849
1.17800	-1.08633	-0.45196	-0.78525	-1.86240	-0.57567
1.19900	-3.65985	-0.41141	-1.18863	-2.81696	-0.47753
1.22000	-1.22962	-0.42014	-2.00296	-4.96665	-0.46434
1.24100	-0.63777	-0.47182	-3.92417	-8.68470	-0.59223
1.26200	-0.47421	-0.57779	-9.37123	-7.66942	-1.19977
1.28300	-0.40976	-0.77225	-12.42227	-4.27713	-1.31742
1.30400	-0.38599	-1.13828	-5.18335	-2.51920	-0.60555
1.32500	-0.38726	-1.89049	-2.70366	-1.65523	-0.43246
1.34600	-0.41014	-3.65139	-1.71887	-1.20233	-0.41115
1.36700	-0.45747	-8.30207	-1.28595	-0.96127	-0.45834
1.38800	-0.53848	-11.48713	-1.11281	-0.86125	-0.57260
1.40900	-0.67278	-5.36219	-1.11003	-0.92418	-0.79345
1.43000	-0.90134	-2.87183	-1.28441	-1.41411	-1.23005
1.45100	-1.31565	-1.84224	-1.75047	-4.06469	-2.18577
1.47200	-2.14157	-1.36932	-2.88640	-3.45894	-4.61817
1.49300	-3.99652	-1.15609	-5.97731	-1.13217	-11.50980
1.51400	-8.45094	-1.10327	-13.79064	-0.61638	-9.15684
1.53500	-10.47921	-1.19997	-7.12740	-0.43592	-4.09591
1.55600	-5.23005	-1.52024	-3.29451	-0.35080	-2.25074
1.57700	-2.84020	-2.33270	-1.88347	-0.30385	-1.45152
1.59800	-1.80026	-4.62291	-1.26815	-0.27685	-1.05640
1.61900	-1.30093	-13.15901	-0.97202	-0.26382	-0.85124
1.64000	-1.05414	-8.90331	-0.83808	-0.26558	-0.76008
1.66100	-0.95612	-3.51125	-0.82311	-0.29413	-0.77159
1.68200	-0.98621	-1.88480	-0.96723	-0.38169	-0.95001
1.70300	-1.20298	-1.26997	-1.53301	-0.68812	-1.61601
1.72400	-1.86922	-1.05088	-4.09901	-2.37099	-4.63229



1.74500	-4.22237	-1.12998	-5.95083	-1.39190	-6.69308
1.76600	-16.50441	-1.89700	-2.41087	-0.52994	-2.02709
1.78700	-5.09161	-5.50433	-2.49015	-0.33305	-0.97641
1.80800	-1.96617	-3.03840	-6.35701	-0.26248	-0.64552
1.82900	-1.08751	-1.20167	-8.15064	-0.22958	-0.53385
1.85000	-0.74116	-0.72529	-2.43722	-0.21188	-0.61340
1.87100	-0.57327	-0.54372	-1.18136	-0.20166	-0.87389
1.89200	-0.48085	-0.45575	-0.75760	-0.19570	-0.46416
1.91300	-0.42599	-0.40709	-0.56753	-0.19246	-0.32911
1.93400	-0.39222	-0.37841	-0.46663	-0.19113	-0.28097
1.95500	-0.37152	-0.36138	-0.40734	-0.19125	-0.25662
1.97600	-0.35962	-0.35195	-0.37038	-0.19256	-0.24199
1.99700	-0.35412	-0.34790	-0.34673	-0.19489	-0.23252
2.01800	-0.35362	-0.34800	-0.33170	-0.19817	-0.22632
2.03900	-0.35731	-0.35151	-0.32266	-0.20234	-0.22242
2.06000	-0.36473	-0.35800	-0.31805	-0.20740	-0.22029
2.08100	-0.37569	-0.36723	-0.31689	-0.21336	-0.21960
2.10200	-0.39020	-0.37910	-0.31858	-0.22026	-0.22015
2.12300	-0.40842	-0.39361	-0.32273	-0.22815	-0.22182
2.14400	-0.43065	-0.41085	-0.32910	-0.23712	-0.22453
2.16500	-0.45740	-0.43098	-0.33755	-0.24727	-0.22823
2.18600	-0.48932	-0.45424	-0.34802	-0.25873	-0.23293
2.20700	-0.52734	-0.48094	-0.36051	-0.27167	-0.23864
2.22800	-0.57266	-0.51150	-0.37508	-0.28627	-0.24538
2.24900	-0.62690	-0.54641	-0.39184	-0.30279	-0.25324
2.27000	-0.69220	-0.58631	-0.41092	-0.32152	-0.26228
2.29100	-0.77146	-0.63198	-0.43253	-0.34285	-0.27262
2.31200	-0.86864	-0.68437	-0.45692	-0.36724	-0.28441
2.33300	-0.98923	-0.74468	-0.48439	-0.39529	-0.29781
2.35400	-1.14096	-0.81436	-0.51533	-0.42776	-0.31306
2.37500	-1.33496	-0.89527	-0.55018	-0.46562	-0.33041
2.39600	-1.58754	-0.98972	-0.58950	-0.51019	-0.35023
2.41700	-1.92315	-1.10061	-0.63396	-0.56316	-0.37293
2.43800	-2.37916	-1.23165	-0.68438	-0.62690	-0.39907
2.45900	-3.01387	-1.38756	-0.74176	-0.70463	-0.42934
2.48000	-3.91930	-1.57434	-0.80731	-0.80095	-0.46465
2.50100	-5.24023	-1.79972	-0.88257	-0.92255	-0.50620
2.52200	-7.18984	-2.07345	-0.96941	-1.07940	-0.55555
2.54300	-9.95124	-2.40781	-1.07020	-1.28696	-0.61487
2.56400	-12.61898	-2.81764	-1.18794	-1.57004	-0.68709
2.58500	-11.71253	-3.31975	-1.32642	-1.97021	-0.77640
2.60600	-8.74968	-3.93021	-1.49052	-2.56072	-0.88880
2.62700	-6.38362	-4.65683	-1.68653	-3.47826	-1.03323
2.64800	-4.77088	-5.48205	-1.92261	-4.99759	-1.22329
2.66900	-3.68185	-6.33074	-2.20936	-7.73938	-1.48049
2.69000	-2.93499	-7.03402	-2.56053	-13.52549	-1.84001
2.71100	-2.41264	-7.35446	-2.99380	-21.51326	-2.36185
2.73200	-2.04049	-7.14982	-3.53121	-10.78491	-3.15264

2.75300	-1.77135	-6.52004	-4.19868	-6.42938	-4.40997
2.77400	-1.57473	-5.70138	-5.02206	-4.27585	-6.51183
2.79500	-1.43058	-4.88297	-6.01442	-3.11689	-10.13006
2.81600	-1.32547	-4.15655	-7.14219	-2.48190	-14.40199
2.83700	-1.25021	-3.54817	-8.25478	-2.15715	-11.14813
2.85800	-1.19850	-3.05320	-9.01700	-2.04477	-7.09968
2.87900	-1.16592	-2.65637	-9.06112	-2.10780	-4.74857
2.90000	-1.14941	-2.34067	-8.39681	-2.34411	-3.38623
2.92100	-1.14689	-2.09084	-7.40044	-2.75866	-2.56811
2.94200	-1.15695	-1.89426	-6.40497	-3.29356	-2.06102
2.96300	-1.17880	-1.74094	-5.55911	-3.71214	-1.74062
2.98400	-1.21204	-1.62313	-4.89759	-3.69873	-1.53840
3.00500	-1.25669	-1.53492	-4.41062	-3.29696	-1.41544
3.02600	-1.31314	-1.47186	-4.07719	-2.83055	-1.34909
3.04700	-1.38212	-1.43067	-3.87774	-2.48870	-1.32616
3.06800	-1.46474	-1.40902	-3.79830	-2.30205	-1.33924
3.08900	-1.56253	-1.40531	-3.83165	-2.25124	-1.38481
3.11000	-1.67752	-1.41860	-3.97756	-2.31709	-1.46220
3.13100	-1.81233	-1.44851	-4.24317	-2.49228	-1.57319
3.15200	-1.97031	-1.49517	-4.64393	-2.78253	-1.72193
3.17300	-2.15577	-1.55919	-5.20552	-3.20682	-1.91524
3.19400	-2.37420	-1.64168	-5.96734	-3.79903	-2.16315
3.21500	-2.63257	-1.74426	-6.98916	-4.61160	-2.47982
3.23600	-2.93973	-1.86913	-8.36441	-5.72104	-2.88479
3.25700	-3.30653	-2.01914	-10.25013	-7.23461	-3.40430
3.27800	-3.74559	-2.19786	-12.94680	-9.29103	-4.07235
3.29900	-4.26961	-2.40982	-17.16721	-12.00975	-4.92925
3.32000	-4.88737	-2.66074	-25.14380	-15.11403	-6.01178
3.34100	-5.59763	-2.95800	-24.51484	-16.71160	-7.31883
3.36200	-6.38870	-3.31121	-17.12025	-15.31881	-8.71908
3.38300	-7.25909	-3.73256	-13.25698	-12.94750	-9.80928
3.40400	-8.25616	-4.23694	-10.85642	-10.65684	-10.05056
3.42500	-9.49503	-4.84241	-9.26331	-8.72362	-9.44134
3.44600	-11.14843	-5.57136	-8.21191	-7.29811	-8.53565
3.46700	-13.45669	-6.45208	-7.59167	-6.32898	-7.76771
3.48800	-16.75901	-7.52002	-7.38213	-5.57933	-7.29657
3.50900	-20.86279	-8.81796	-7.61925	-4.88264	-7.13758
3.53000	-20.71563	-10.39120	-8.33526	-4.29853	-7.23622
3.55100	-16.65282	-12.26617	-9.39178	-3.89553	-7.47163
3.57200	-13.51509	-14.38322	-10.26084	-3.67065	-7.65544
3.59300	-11.44670	-16.47098	-10.43969	-3.58041	-7.61853
3.61400	-10.16654	-18.11291	-10.26010	-3.54896	-7.37105
3.63500	-9.51774	-19.38055	-10.23344	-3.54235	-7.11966
3.65600	-9.43395	-20.84181	-10.22722	-3.69848	-7.11984
3.67700	-9.90996	-21.81083	-9.70870	-4.30222	-7.58062
3.69800	-10.99601	-20.05169	-8.77638	-5.76562	-8.63691
3.71900	-12.81898	-17.07110	-8.12482	-8.37246	-10.17274
3.74000	-15.65207	-14.71141	-8.26650	-9.55359	-11.30063

3.76100	-20.06486	-13.21724	-9.14100	-7.17550	-11.04890
3.78200	-25.90431	-12.53520	-8.85390	-5.08856	-10.47191
3.80300	-23.75829	-12.44374	-6.90177	-3.87805	-10.97605
3.82400	-19.22754	-11.96893	-5.29743	-3.18090	-13.03510
3.84500	-15.74014	-10.24905	-4.31615	-2.76226	-14.07849
3.86600	-12.66623	-8.32929	-3.71306	-2.50258	-11.13944
3.88700	-10.25923	-6.92258	-3.32206	-2.34048	-8.31852
3.90800	-8.53874	-5.96796	-3.05649	-2.24299	-6.50236
3.92900	-7.31438	-5.29990	-2.87089	-2.19160	-5.33620
3.95000	-6.41720	-4.80944	-2.74016	-2.17558	-4.55466
3.97100	-5.73639	-4.43475	-2.64970	-2.18874	-4.00949
3.99200	-5.20433	-4.14047	-2.59063	-2.22762	-3.61778
4.01300	-4.77963	-3.90540	-2.55745	-2.29058	-3.33119
4.03400	-4.43596	-3.71616	-2.54671	-2.37733	-3.12026
4.05500	-4.15583	-3.56392	-2.55637	-2.48859	-2.96631
4.07600	-3.92713	-3.44260	-2.58534	-2.62601	-2.85713
4.09700	-3.74125	-3.34796	-2.63330	-2.79206	-2.78464
4.11800	-3.59195	-3.27700	-2.70055	-2.99007	-2.74342
4.13900	-3.47468	-3.22760	-2.78793	-3.22425	-2.72995
4.16000	-3.38614	-3.19838	-2.89682	-3.49976	-2.74208
4.18100	-3.32402	-3.18847	-3.02914	-3.82304	-2.77870
4.20200	-3.28683	-3.19750	-3.18739	-4.20284	-2.83957
4.22300	-3.27376	-3.22555	-3.37472	-4.65226	-2.92522
4.24400	-3.28462	-3.27309	-3.59502	-5.19069	-3.03688
4.26500	-3.31984	-3.34103	-3.85303	-5.84266	-3.17646
4.28600	-3.38044	-3.43072	-4.15449	-6.63581	-3.34663
4.30700	-3.46806	-3.54401	-4.50633	-7.60334	-3.55082
4.32800	-3.58499	-3.68332	-4.91687	-8.78994	-3.79335
4.34900	-3.73416	-3.85163	-5.39608	-10.25762	-4.07948
4.37000	-3.91904	-4.05250	-5.95590	-12.08768	-4.41557
4.39100	-4.14315	-4.28975	-6.61046	-14.35207	-4.80910
4.41200	-4.40907	-4.56660	-7.37632	-16.89842	-5.26877
4.43300	-4.71656	-4.88428	-8.27229	-18.51660	-5.80453
4.45400	-5.06149	-5.24153	-9.31829	-17.45814	-6.42760
4.47500	-5.43919	-5.63796	-10.53151	-15.10394	-7.15007
4.49600	-5.85324	-6.07981	-11.91476	-13.09983	-7.98340
4.51700	-6.31907	-6.57977	-13.42526	-11.69679	-8.93505
4.53800	-6.85472	-7.15071	-14.90497	-10.52251	-10.00140
4.55900	-7.47017	-7.80091	-15.99764	-9.34831	-11.15400
4.58000	-8.16291	-8.53212	-16.27735	-8.26028	-12.31806
4.60100	-8.91488	-9.33653	-15.71258	-7.33091	-13.35550
4.62200	-9.68583	-10.19011	-14.71996	-6.56324	-14.09563
4.64300	-10.40414	-11.04175	-13.70106	-5.93657	-14.43068
4.66400	-10.96479	-11.80187	-12.86288	-5.42854	-14.31854
4.68500	-11.25396	-12.34527	-12.29489	-5.02088	-13.59839
4.70600	-11.20860	-12.55099	-12.04969	-4.70004	-12.18702
4.72700	-10.86693	-12.37290	-12.18360	-4.45668	-10.51890
4.74800	-10.34143	-11.87507	-12.76841	-4.28535	-9.04344

4.76900	-9.73631	-11.19081	-13.85148	-4.18471	-7.86736
4.79000	-9.11182	-10.45892	-15.22063	-4.15921	-6.94116
4.81100	-8.50344	-9.78746	-15.79344	-4.22317	-6.19980
4.83200	-7.93562	-9.24122	-14.40325	-4.40466	-5.59481
4.85300	-7.42317	-8.84520	-12.10796	-4.74527	-5.09333
4.87400	-6.97272	-8.60226	-10.08180	-5.30245	-4.67295
4.89500	-6.58555	-8.51109	-8.57104	-6.16523	-4.31788
4.91600	-6.26011	-8.57271	-7.51484	-7.48403	-4.01662
4.93700	-5.99383	-8.78467	-6.81947	-9.52181	-3.76059
4.95800	-5.78420	-9.12262	-6.40944	-12.74160	-3.54320
4.97900	-5.62951	-9.50582	-6.21862	-17.42303	-3.35940
5.00000	-5.52964	-9.76138	-6.20433	-17.54434	-3.20529

## APPENDIX C: VHDL CODE

VHDL codes are generated automatically from Matlab Simulink NN-antenna model using Xilinx System Generator. The VHDL for filtenna antenna is presented here as an example for generated VHDL code.

```
-----  
-- System Generator version 12.1 VHDL source file.  
-- Copyright(C) 2010 by Xilinx, Inc. All rights reserved.  
-----  
  
library IEEE;  
use IEEE.std_logic_1164.all;  
use IEEE.numeric_std.all;  
use work.conv_pkg.all;  
-- synopsys translate_off  
library unisim;  
use unisim.vcomponents.all;  
-- synopsys translate_on  
entity xlclockdriver is  
  generic (  
    period: integer := 2;  
    log_2_period: integer := 0;  
    pipeline_regs: integer := 5;  
    use_bufg: integer := 0  
  );  
  port (  

```

```

sysclk: in std_logic;
sysclr: in std_logic;
sysce: in std_logic;
clk: out std_logic;
clr: out std_logic;
ce: out std_logic;
ce_logic: out std_logic
);
end xclockdriver;

```

architecture behavior of xclockdriver is

```

component bufg

```

```

  port (
    i: in std_logic;
    o: out std_logic
  );

```

```

end component;

```

```

component synth_reg_w_init

```

```

  generic (
    width: integer;
    init_index: integer;
    init_value: bit_vector;
    latency: integer
  );

```

```

  port (
    i: in std_logic_vector(width - 1 downto 0);
    ce: in std_logic;
    clr: in std_logic;
    clk: in std_logic;
  );

```

```

    o: out std_logic_vector(width - 1 downto 0)
);
end component;

function size_of_uint(inp: integer; power_of_2: boolean)
    return integer
is
    constant inp_vec: std_logic_vector(31 downto 0) :=
        integer_to_std_logic_vector(inp,32, xUnsigned);
    variable result: integer;
begin
    result := 32;
    for i in 0 to 31 loop
        if inp_vec(i) = '1' then
            result := i;
        end if;
    end loop;
    if power_of_2 then
        return result;
    else
        return result+1;
    end if;
end;

function is_power_of_2(inp: std_logic_vector)
    return boolean
is
    constant width: integer := inp'length;
    variable vec: std_logic_vector(width - 1 downto 0);
    variable single_bit_set: boolean;

```

```

variable more_than_one_bit_set: boolean;

variable result: boolean;

begin

vec := inp;

single_bit_set := false;

more_than_one_bit_set := false;

-- synopsys translate_off

if (is_XorU(vec)) then

    return false;

end if;

-- synopsys translate_on

if width > 0 then

    for i in 0 to width - 1 loop

        if vec(i) = '1' then

            if single_bit_set then

                more_than_one_bit_set := true;

            end if;

            single_bit_set := true;

        end if;

    end loop;

end if;

if (single_bit_set and not(more_than_one_bit_set)) then

    result := true;

else

    result := false;

end if;

return result;

end;

```



```

function ce_reg_init_val(index, period : integer)
    return integer
is
    variable result: integer;
begin
    result := 0;
    if ((index mod period) = 0) then
        result := 1;
    end if;
    return result;
end;

function remaining_pipe_regs(num_pipeline_regs, period : integer)
    return integer
is
    variable factor, result: integer;
begin
    factor := (num_pipeline_regs / period);
    result := num_pipeline_regs - (period * factor) + 1;
    return result;
end;

function sg_min(L, R: INTEGER) return INTEGER is
begin
    if L < R then
        return L;
    else
        return R;
    end if;
end;

```

```

end;

constant max_pipeline_regs : integer := 8;

constant pipe_regs : integer := 5;

constant num_pipeline_regs : integer := sg_min(pipeline_regs, max_pipeline_regs);

constant rem_pipeline_regs : integer := remaining_pipe_regs(num_pipeline_regs,period);

constant period_floor: integer := max(2, period);

constant power_of_2_counter: boolean :=
    is_power_of_2(integer_to_std_logic_vector(period_floor,32, xUnsigned));

constant cnt_width: integer :=
    size_of_uint(period_floor, power_of_2_counter);

constant clk_for_ce_pulse_minus1: std_logic_vector(cnt_width - 1 downto 0) :=
    integer_to_std_logic_vector((period_floor - 2),cnt_width, xUnsigned);

constant clk_for_ce_pulse_minus2: std_logic_vector(cnt_width - 1 downto 0) :=
    integer_to_std_logic_vector(max(0,period - 3),cnt_width, xUnsigned);

constant clk_for_ce_pulse_minus_regs: std_logic_vector(cnt_width - 1 downto 0) :=
    integer_to_std_logic_vector(max(0,period - rem_pipeline_regs),cnt_width, xUnsigned);

signal clk_num: unsigned(cnt_width - 1 downto 0) := (others => '0');

signal ce_vec : std_logic_vector(num_pipeline_regs downto 0);

attribute MAX_FANOUT : string;

attribute MAX_FANOUT of ce_vec:signal is "REDUCE";

signal ce_vec_logic : std_logic_vector(num_pipeline_regs downto 0);

attribute MAX_FANOUT of ce_vec_logic:signal is "REDUCE";

signal internal_ce: std_logic_vector(0 downto 0);

signal internal_ce_logic: std_logic_vector(0 downto 0);

signal cnt_clr, cnt_clr_dly: std_logic_vector (0 downto 0);

begin

clk <= sysclk;

clr <= sysclr;

```

```

cntr_gen: process(sysclk)
begin
  if sysclk'event and sysclk = '1' then
    if (sysce = '1') then
      if ((cnt_clr_dly(0) = '1') or (sysclr = '1')) then
        clk_num <= (others => '0');
      else
        clk_num <= clk_num + 1;
      end if;
    end if;
  end if;
end process;

clr_gen: process(clk_num, sysclr)
begin
  if power_of_2_counter then
    cnt_clr(0) <= sysclr;
  else
    if (unsigned_to_std_logic_vector(clk_num) = clk_for_ce_pulse_minus1
      or sysclr = '1') then
      cnt_clr(0) <= '1';
    else
      cnt_clr(0) <= '0';
    end if;
  end if;
end process;

clr_reg: synth_reg_w_init
generic map (
  width => 1,

```

```

init_index => 0,
init_value => b"0000",
latency => 1
)
port map (
i => cnt_clr,
ce => sysce,
clr => sysclr,
clk => sysclk,
o => cnt_clr_dly
);
pipelined_ce : if period > 1 generate
ce_gen: process(clk_num)
begin
if unsigned_to_std_logic_vector(clk_num) = clk_for_ce_pulse_minus_regs then
ce_vec(num_pipeline_regs) <= '1';
else
ce_vec(num_pipeline_regs) <= '0';
end if;
end process;
ce_pipeline: for index in num_pipeline_regs downto 1 generate
ce_reg : synth_reg_w_init
generic map (
width => 1,
init_index => ce_reg_init_val(index, period),
init_value => b"0000",
latency => 1
)

```

```

port map (
    i => ce_vec(index downto index),
    ce => sysce,
    clr => sysclr,
    clk => sysclk,
    o => ce_vec(index-1 downto index-1)
);

end generate;

internal_ce <= ce_vec(0 downto 0);

end generate;

pipelined_ce_logic: if period > 1 generate
    ce_gen_logic: process(clk_num)
    begin
        if unsigned_to_std_logic_vector(clk_num) = clk_for_ce_pulse_minus_regs then
            ce_vec_logic(num_pipeline_regs) <= '1';
        else
            ce_vec_logic(num_pipeline_regs) <= '0';
        end if;
    end process;

ce_logic_pipeline: for index in num_pipeline_regs downto 1 generate
    ce_logic_reg : synth_reg_w_init
        generic map (
            width => 1,
            init_index => ce_reg_init_val(index, period),
            init_value => b"0000",
            latency => 1
        )
    port map (

```

```

        i => ce_vec_logic(index downto index),
        ce => sysce,
        clr => sysclr,
        clk => sysclk,
        o => ce_vec_logic(index-1 downto index-1)
    );
end generate;

internal_ce_logic <= ce_vec_logic(0 downto 0);
end generate;

use_bufg_true: if period > 1 and use_bufg = 1 generate
    ce_bufg_inst: bufg
        port map (
            i => internal_ce(0),
            o => ce
        );
    ce_bufg_inst_logic: bufg
        port map (
            i => internal_ce_logic(0),
            o => ce_logic
        );
end generate;

use_bufg_false: if period > 1 and (use_bufg = 0) generate
    ce <= internal_ce(0);
    ce_logic <= internal_ce_logic(0);
end generate;

generate_system_clk: if period = 1 generate
    ce <= sysce;
    ce_logic <= sysce;
end generate;

```

```

end generate;

end architecture behavior;

library IEEE;

use IEEE.std_logic_1164.all;

use work.conv_pkg.all;

entity default_clock_driver is

port (

sysce: in std_logic;

sysce_clr: in std_logic;

sysclk: in std_logic;

ce_1: out std_logic;

clk_1: out std_logic

);

end default_clock_driver;

architecture structural of default_clock_driver is

attribute syn_noprune: boolean;

attribute syn_noprune of structural : architecture is true;

attribute optimize_primitives: boolean;

attribute optimize_primitives of structural : architecture is false;

attribute dont_touch: boolean;

attribute dont_touch of structural : architecture is true;

signal sysce_clr_x0: std_logic;

signal sysce_x0: std_logic;

signal sysclk_x0: std_logic;

signal xlclockdriver_1_ce: std_logic;

```

```
signal xlclockdriver_1_clk: std_logic;
```

```
begin
```

```
sysce_x0 <= sysce;
```

```
sysce_clr_x0 <= sysce_clr;
```

```
sysclk_x0 <= sysclk;
```

```
ce_1 <= xlclockdriver_1_ce;
```

```
clk_1 <= xlclockdriver_1_clk;
```

```
xlclockdriver_1: entity work.xlclockdriver
```

```
generic map (
```

```
log_2_period => 1,
```

```
period => 1,
```

```
use_bufg => 0
```

```
)
```

```
port map (
```

```
sysce => sysce_x0,
```

```
sysclk => sysclk_x0,
```

```
sysclr => sysce_clr_x0,
```

```
ce => xlclockdriver_1_ce,
```

```
clk => xlclockdriver_1_clk
```

```
);
```

```
end structural;
```

```
library IEEE;
```

```
use IEEE.std_logic_1164.all;
```

```
use work.conv_pkg.all;
```



```

entity filtenna_final_3_cw is
  port (
    ce: in std_logic := '1';
    clk: in std_logic; -- clock period = 500000.0 ns (0.0020 Mhz)
    gateway_in: in std_logic_vector(15 downto 0);
    gateway_out: out std_logic_vector(15 downto 0)
  );
end filtenna_final_3_cw;

```

architecture structural of filtenna\_final\_3\_cw is

```

component xlpersistentdff
  port (
    clk: in std_logic;
    d: in std_logic;
    q: out std_logic
  );
end component;

attribute syn_black_box: boolean;
attribute syn_black_box of xlpersistentdff: component is true;
attribute box_type: string;
attribute box_type of xlpersistentdff: component is "black_box";
attribute syn_noprune: boolean;
attribute optimize_primitives: boolean;
attribute dont_touch: boolean;
attribute syn_noprune of xlpersistentdff: component is true;
attribute optimize_primitives of xlpersistentdff: component is false;
attribute dont_touch of xlpersistentdff: component is true;

```

```

signal ce_1_sg_x27: std_logic;
attribute MAX_FANOUT: string;
attribute MAX_FANOUT of ce_1_sg_x27: signal is "REDUCE";
signal clkNet: std_logic;
signal clk_1_sg_x27: std_logic;
signal gateway_in_net: std_logic_vector(15 downto 0);
signal gateway_out_net: std_logic_vector(15 downto 0);
signal persistentdff_inst_q: std_logic;
attribute syn_keep: boolean;
attribute syn_keep of persistentdff_inst_q: signal is true;
attribute keep: boolean;
attribute keep of persistentdff_inst_q: signal is true;
attribute preserve_signal: boolean;
attribute preserve_signal of persistentdff_inst_q: signal is true;

begin

clkNet <= clk;

gateway_in_net <= gateway_in;
gateway_out <= gateway_out_net;

default_clock_driver_x0: entity work.default_clock_driver
port map (
sysce => '1',
sysce_clr => '0',
sysclk => clkNet,
ce_1 => ce_1_sg_x27,
clk_1 => clk_1_sg_x27
);

```

```
filtenna_final_3_x0: entity work.filtenna_final_3
```

```
port map (
```

```
ce_1 => ce_1_sg_x27,
```

```
clk_1 => clk_1_sg_x27,
```

```
gateway_in => gateway_in_net,
```

```
gateway_out => gateway_out_net
```

```
);
```

```
persistendff_inst: xlpersistendff
```

```
port map (
```

```
clk => clkNet,
```

```
d => persistendff_inst_q,
```

```
q => persistendff_inst_q
```

```
);
```

```
end structural;
```

## APPENDIX D: SIMULATION MODELS

Matlab .m codes shown in appendix A generate a Matlab Simulink model for the neural network. This model then needs to be rebuilt using Xilinx System Generator (XSG) blocks. Here, XSG Simulink model for the filtenna antenna is shown in details.

Figure (C.1) shows the outside bocks for the model, Layer 1 represents the hidden layer, while Layer 2 represents the output layer. Subsystem and Subsystem1 represent pre-processing and post-processing of the signals.

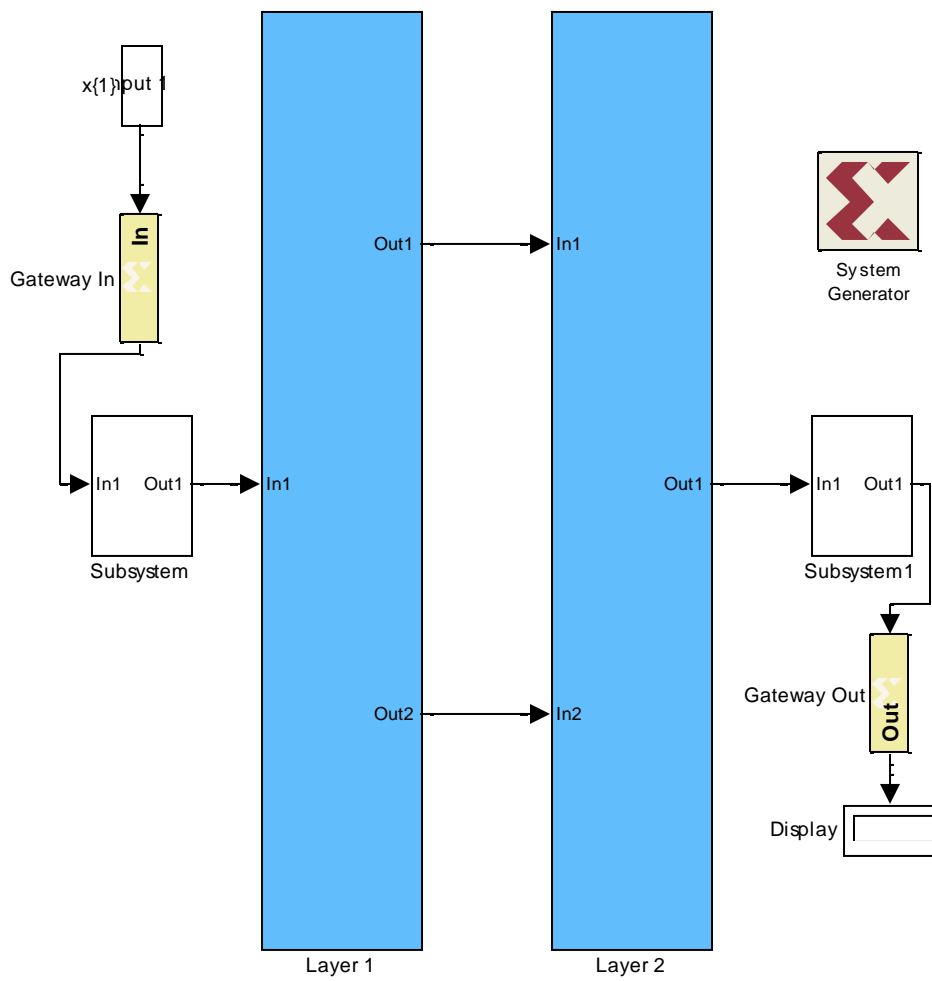


Fig. C.1: Simulink NN\_FPGA model of the antenna

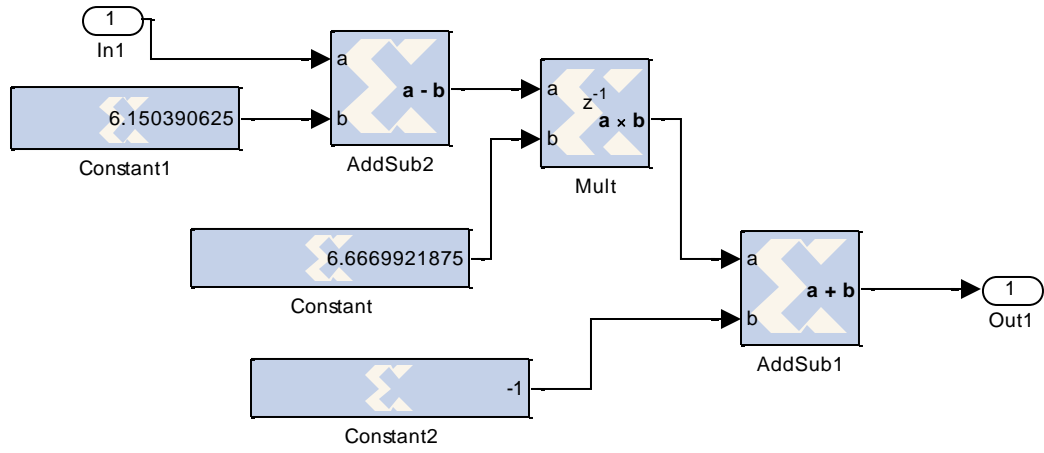


Fig. C.2: Pre-processing

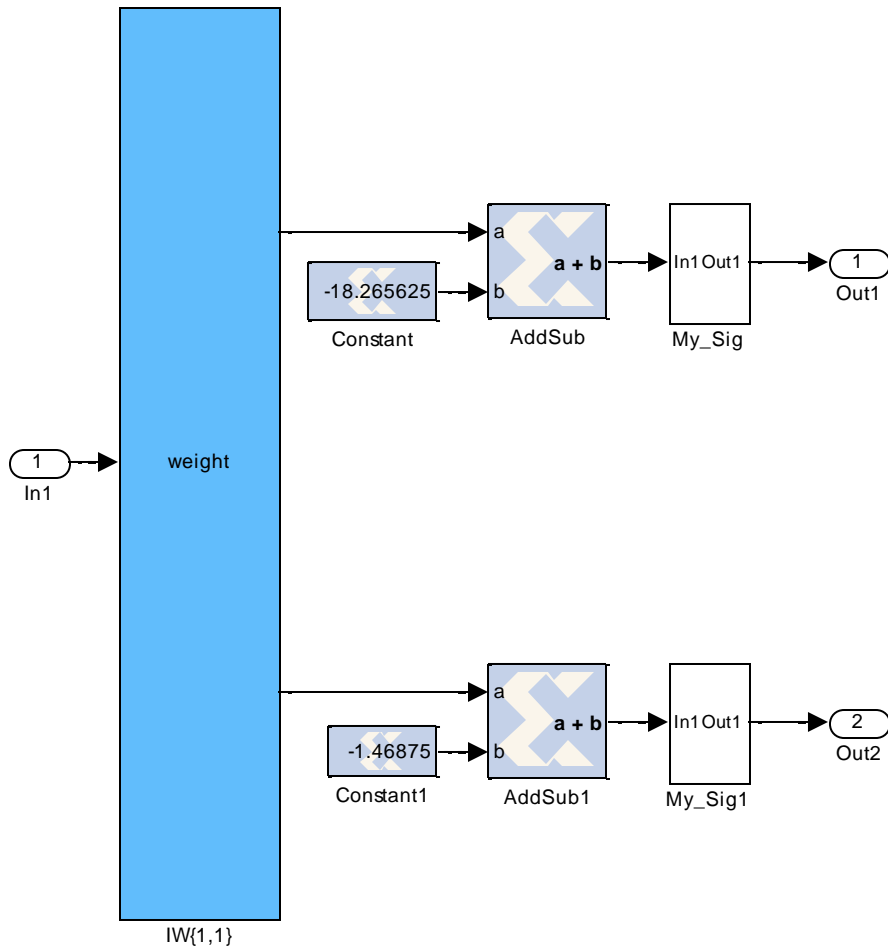


Fig. C.3: Hidden layer of filtenna antenna

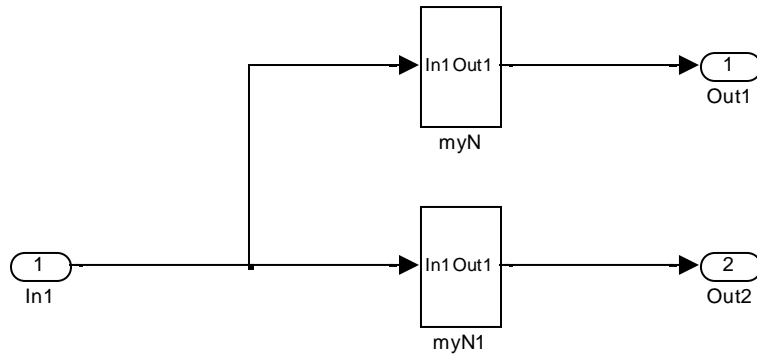


Fig. C.3: Hidden neurons

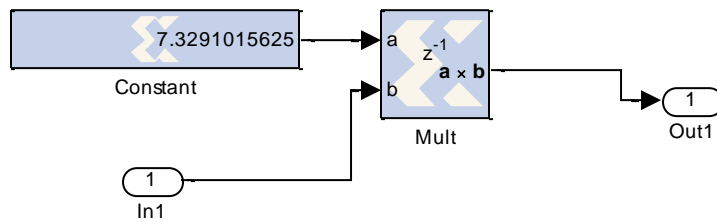


Fig. C.4: Hidden neuron 1

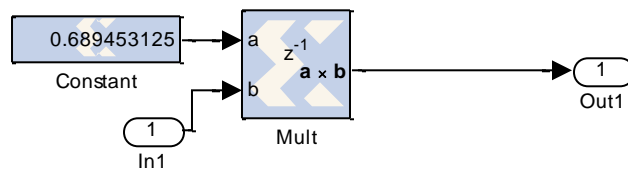


Fig. C.5: Hidden neuron 2



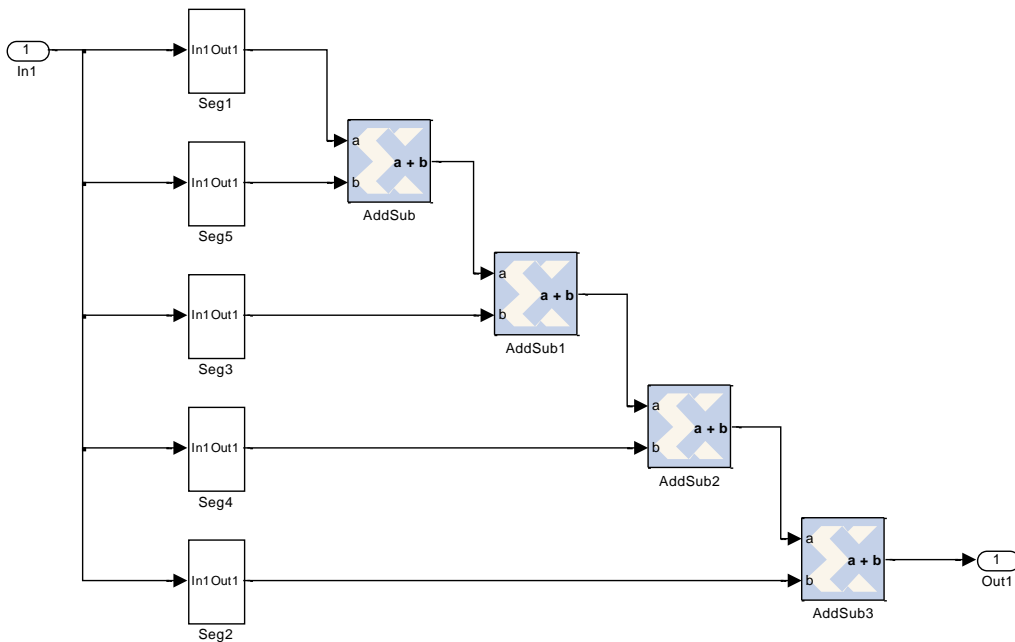


Fig. C.6: Sigmoid function

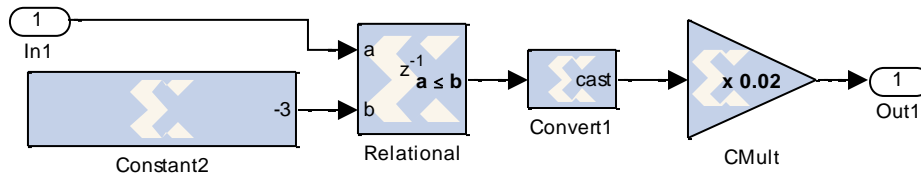


Fig. C.7: Sigmoid function Segment 1

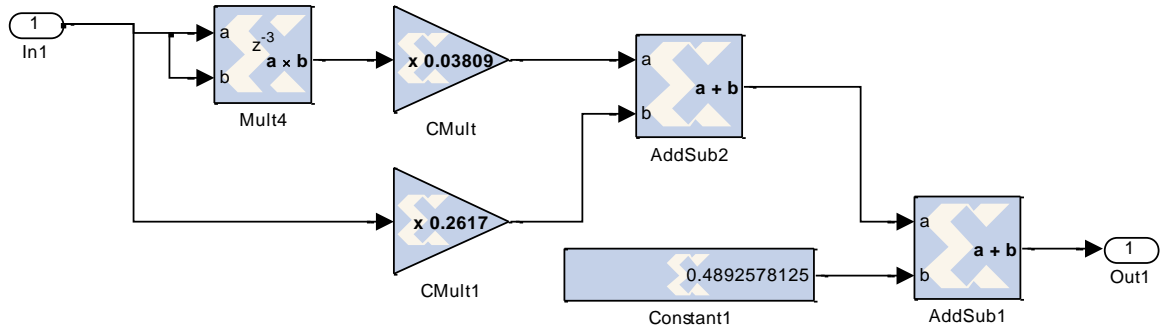


Fig. C.8: Sigmoid function Segment 2

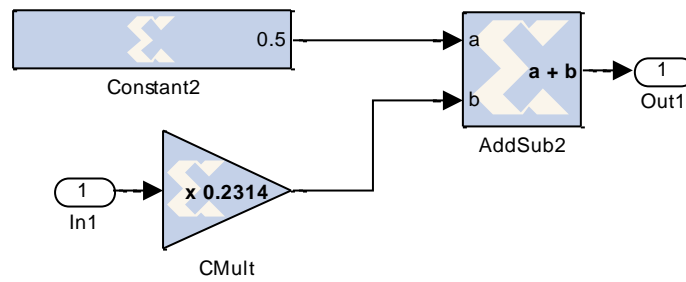


Fig. C.9: Sigmoid function Segment 3 function

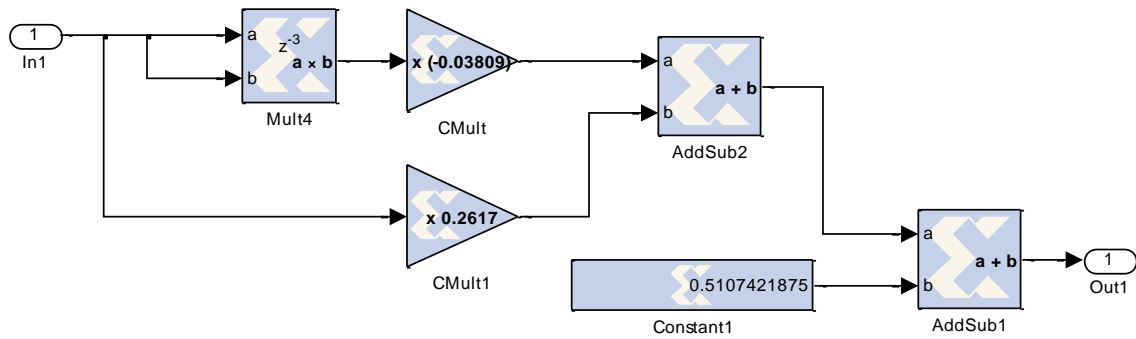


Fig. C.10: Sigmoid function Segment 4 function

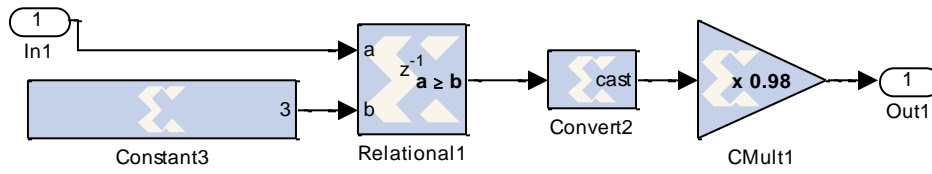


Fig. C.11: Sigmoid function Segment 5

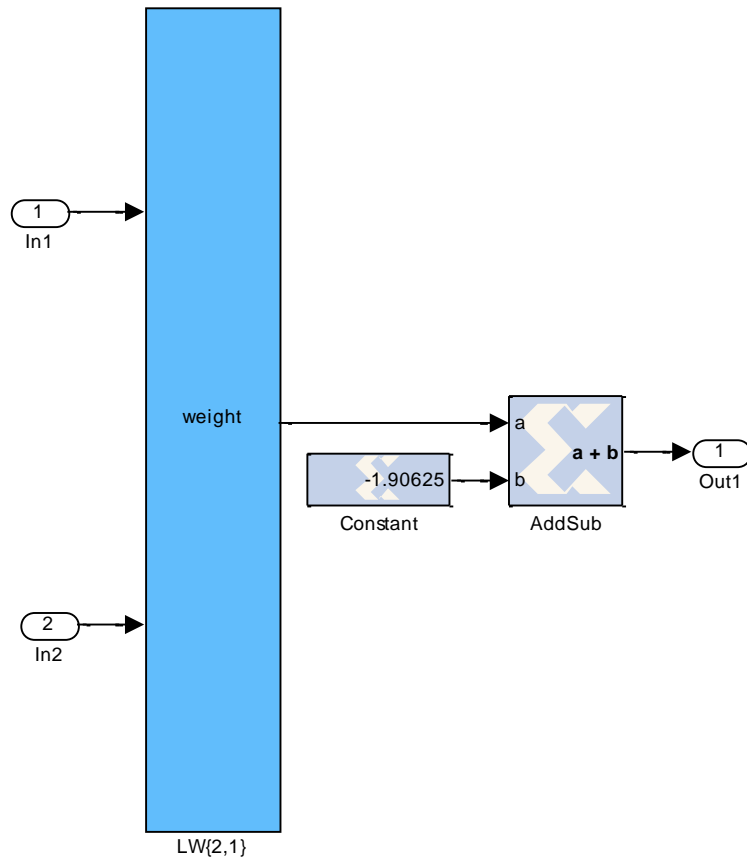


Fig. C.12: Output layer of filtenna antenna

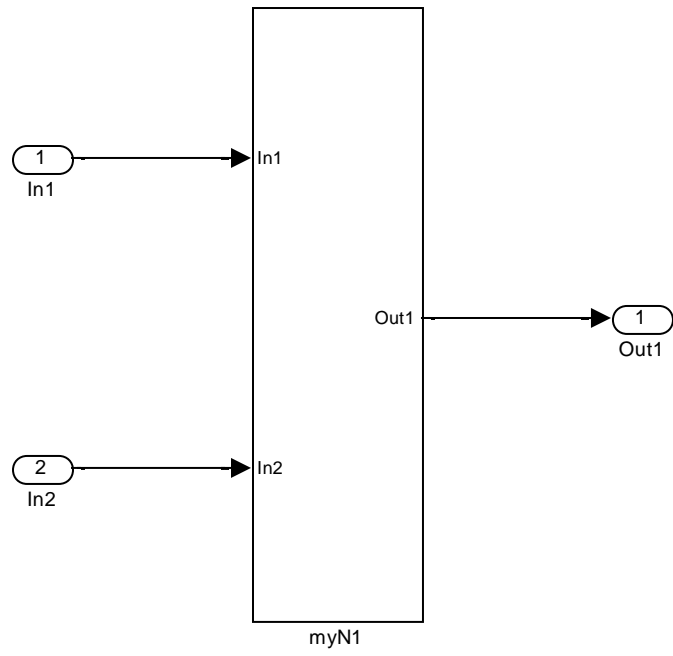


Fig. C.13: Output neuron

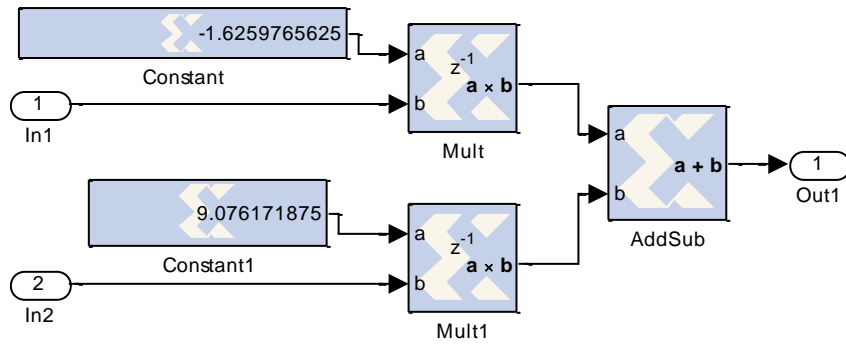


Fig. C.14: output neuron structure

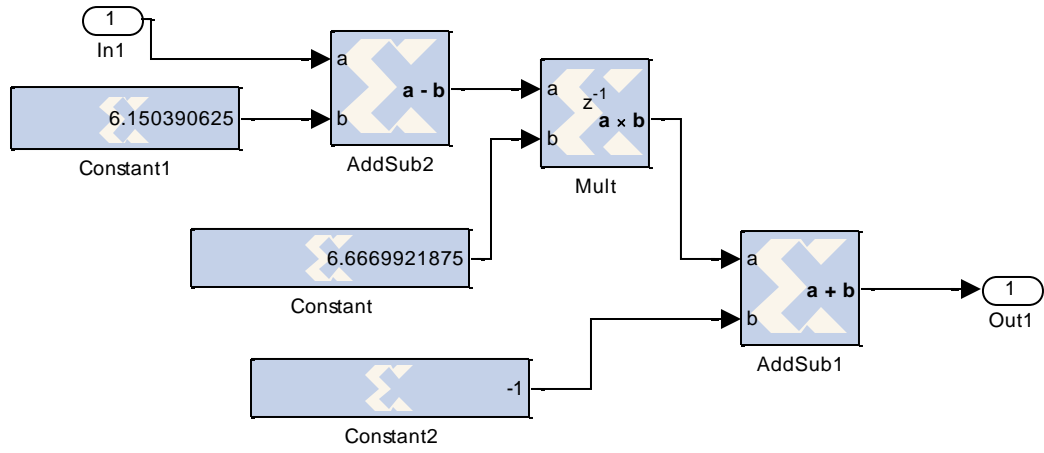


Fig. C.15: Post-processing

## REFERENCES

- [1] Rao M, Srinivas J., *Neural Networks Algorithms and Applications*. London: Alpha Science; 2003.
- [2] Capizzi G, Coco S, Giuffrida C, Laudani A. A Neural Network Approach for the Differentiation of Numerical Solutions of 3-D Electromagnetic Problems. *IEEE Transactions on Magnetics* 2004; 40(2): 953-956.
- [3] Zooghby A, Christodoulou C, Georgiopoulos M. Neural Network-Based Adaptive Beamforming for One- and Two-Dimensional Antenna Arrays. *IEEE Transactions on Antennas and Propagation* 1998; 46(12):1891-1893.
- [4] Patnaik A, Anagnostou D, Christodoulou C, Lyke J. Neurocomputational Analysis of a Multiband Reconfigurable Planar Antenna. *IEEE Transactions on Antennas and Propagation* 2005; 53(11): 3453-3458.
- [5] Patnaik A, Anagnostou D, Christodoulou C, Lyke J. A Frequency Reconfigurable Antenna Design using Neural Network. *IEEE Antennas and Propagation* 2005; 2(A): 409-412.
- [6] Lagaris I, Likas A, Papageorgiou D. Neural-Network Methods for Boundary Value Problems with Irregular Boundaries. *IEEE Transactions on Neural Networks* 2000; 11(5): 1041-1049.
- [7] Christodoulou C, Georgiopoulos M. "Neural Network Applications in Electromagnetics. Boston: Artech House; 2001.
- [8] Ziemianski L. Hybrid Neural Network/Finite Element Modeling of Wave Propagation in Infinite Domains. *Computers and Structures* 2003; 81: 1099-1109.
- [9] Q. J. Zhang, K. C. Gupta, "Neural Networks for RF and Microwave Design", Artech House, 2000.
- [10] Po-Rong Chang; Wen-Hao Yang; Kuan-Kin Chan; , "A neural network approach to MVDR beamforming problem," *Antennas and Propagation, IEEE Transactions on* , vol.40, no.3, pp.313-322, Mar 1992
- [11] J. Lyke, "A Cellular Automata FPGA Architecture that can be Trained with Neural Networks", *Aerospace Conference Proceedings*, Vol. 5, pp 2347-2354, 2002.

- [12] M. Bonnici, E. J. Gatt, J. Micallef, I. Grech, "Artificial Neural Network Optimization for FPGA", 13th IEEE International Conference on Electronics, Circuits and Systems, pp 1340-1343, Dec 2006.
- [13] J. L. Bastos, H. P. Figueroa, A. Monti, "FPGA Implementation of Neural Network-Based Controllers for Power Electronics Applications", Applied Power Electronics Conference and Exposition, pp 1443-1484, March 2006.
- [14] Bernander O. Neural Network. Microsoft Encarta 2006, Redmond, WA, Microsoft Corporation; 2005.
- [15] Haykin S., Neural Networks: A Comprehensive Foundation, 2nd edition, Prentice Hall, 1998.
- [16] Jennifer T. Bernhard, Reconfigurable Antennas (Synthesis Lectures on Antennas and Propagation), 1st edition, Morgan & Claypool Publishers, 2006.
- [17] Constantine A. Balanis, Antenna Theory: Analysis and Design, 3rd Edition, Wiley-Interscience, 2005.
- [18] K. Chung, Y. Nam, T. Yun, and J. Choi, "Reconfigurable Microstrip Patch Antenna with Switchable Polarization", ETRI Journal, Volume 28, Number 3, June 2006
- [19] H. Y. Yang and H. G. Alexopoulous, "Gain enhancement methods for printed circuit antennas through multiple superstrates," IEEE Trans. of Antennas and Propagat., vol.AP-35, pp.860-863, 1987.
- [20] R. Q. Lee and K. F. Lee, "Experimental Study of Two-layer Electromagnetically Coupled Rectangular Patch Antenna," IEEE Trans. Of Antennas and Propagat., vol.AP-38, no.8, pp.1298-1302, Aug. 1990.
- [21] Design of a compact microstrip antenna at 2.4 GHz by Nurulrodziah Bt Abdul Ghafar, University of Technology Malaysia, November 2005.
- [22] K. Araki, H. Ueda and M. Takahashi, "Numerical Analysis of Circular Disk Microstrip Antenna with Parasitic Elements," IEEE Trans. of Antennas and Propagat., vol.AP-34, no.12, pp.1390-1394, Dec. 1986.
- [23] E. Nishiyama, M. Aikawa and S. Egashira, "Three-Element Stacked Microstrip Antenna with Wide-Band and High-Gain Performances," 2003 Int. IEEE/AP-S Symp. Dig., v01.2, pp.900-903, June. 2003.



- [24] R. B. Waterhouse, "Small microstrip patch antenna," *Electron. Letters*, vol. 31, pp. 604–605, Apr. 1995.
- [25] Pong P. Chu, *RTL Hardware Design Using VHDL*, John Wiley and Sons Inc., Hoboken, NJ, 2006.
- [26] IEEE Standard Test Access Port and Boundary-Scan Architecture, [http://standards.ieee.org/reading/ieee/std\\_public/description/testtech/1149.1-1990\\_desc.html](http://standards.ieee.org/reading/ieee/std_public/description/testtech/1149.1-1990_desc.html)
- [27] Neil H. E. Weste and D. Harris, *CMOS VLSI Design*, Pearson Education, Boston, MA, 2005.
- [28] Renovell, M.; Faure, P.; Portal, J.M.; Figueras, J.; Zorian, Y.; , "IS-FPGA : a new symmetric FPGA architecture with implicit scan," *Test Conference*, 2001. *Proceedings. International* , vol., no., pp.924-931, 2001
- [29] Djalal Eddine , Aissa Kheldoun, Larbi Refoufi," Sigmoid function approximation for ANN implementation in FPGA devices", *CSECS '10 Proceedings of the 9th WSEAS international conference on Circuits, systems, electronics, control & signal processing*, 2010.
- [30] Y. Tawk, J. Costantine, and C. G. Christodoulou, "A Varactor Based Reconfigurable Filtenna", *IEEE Antennas and Wireless Propagation Letters*.
- [31] S. Nikolaou, G. E. Ponchak, J. Papapolymerou, and M. M. Tentzeris, "Conformal double exponentially tapered slot antenna (DETTSA) on LCP for UWB applications," *IEEE Transactions on Antennas and Propagation*, vol.54, no.6, pp. 1663- 1669, June 2006.
- [32] Alayesh, M.A.; Christodoulou, C.G.; Joler, M.; Barbin, S.E.; , "Reconfigurable multi-band stacked Microstrip Patch Antenna for wireless applications," *Antennas and Propagation Conference*, 2008. *LAPC 2008. Loughborough* , vol., no., pp.329-332, 17-18 March 2008.
- [33] Tawk, Y.; Costantine, J.; Christodoulou, C.G.; , "A frequency reconfigurable rotatable microstrip antenna design," *Antennas and Propagation Society International Symposium (APSURSI)*, 2010 *IEEE* , vol., no., pp.1-4, 11-17 July 2010
- [34] Al-Husseini, M.; Ramadan, A.; Tawk, Y.; Christodoulou, C.; Kabalan, K.; El-Hajj, A.; , "A planar ultrawideband antenna with multiple controllable band notches

for UWB cognitive radio applications," *Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference on*, vol., no., pp.375-377, 11-15 April 2011.

[35] M. E. Zamudio, J. -H. Kim, Y. Tawk and C. G. Christodoulou," Integrated Cognitive Radio Antenna Using Reconfigurable Band Pass Filters", *European Conference on Antennas and Propagation 2011*, 11-15 April 2011, Rome.

[36] J. Costantine, C. G. Christodoulou, S. E. Barbin, "A New Reconfigurable Multi Band Patch Antenna", *IEEE IMOC Conference, Salvador, Brazil*, pp.75-78, October 2007.