ISSN 2320 2599
Volume 2, No.6, November - December 2013
International Journal of Microwaves Applications
Available Online at http://warse.org/pdfs/2013/ijma03262013.pdf

Design and Analysis of L-slotted Microstrip Patch Antenna with Different Artificial Neural Network models

Moumita Guha[1], Pritam Singha Roy[2] and Samik Chakraborty[3]
[1]Research Scholar, Department of IEE, Jadavpur University,Kolkata,India,mou.guha08@gmail.com
[2]Assistant Professor, Department of AEIE,DIET,Murshidabad,India,prittam.pritam@gmail.com
[3]Professor, Department of ECE, Indian Maritime University, Kolkata, India

ABSTRACT

This paper, different neural network models has been used to estimation of resonance frequency of a coaxial feed L-shaped Microstrip patch Antenna. The 6 Different training algorithms of Multi-Layer Feed forward back Propagation (MLFFBP) and RBF network have been used to implement the neural network model. The aim of the study is determine of physical parameters without any mathematical expressions. Compared the results of different neural network for MLFFBP training algorithm and Radial Basis Function (RBF). Number of neurons and number of hidden layer is also carried out for estimating the resonance frequency. The method of moment (MOM) based IE3D software was used to generate data dictionary for training and validation set of ANN. The results obtain using ANN are compared with simulation feeding and found quite satisfactory.

Key words: Artificial Neural Network, L-slot, Microstrip Antenna, Radial Basis function, Resonance frequency.

1. INTRODUCTION

Microstrip antennas due to their many attractive features have drawn attention of industries for an ultimate solution for wireless communication. The existing era of wireless communication has led to the design of an efficient, wide band, low cost and small volume antennas which can readily be incorporated into a broad spectrum of systems [1, 2].sufficient amount of work [3-10] indicates how ANN have been used efficiently to design rectangular Microstrip antenna for the determination of different patch dimensions i.e. length, width, resonant frequency, radiation efficiency etc.

In this paper, an attempt has been made to exploit the capability of artificial neural networks to calculate the resonating frequency, bandwidth and gain of coaxial feed rectangular Microstrip patch antenna. The trained ANN is used to determine different important antenna characteristics for various structural input variables. Neuro models are computationally much more efficient than EM models once they are trained with reliable learning data obtained from a "fine" model by either EM simulation or measurement [3, 4, 5, 6].The neuro models can be used for efficient and accurate optimization and design within the range of training. Electromagnetic simulations which are based on full –wave analysis techniques, such as Method of Moment (used in IE3D^{TM}) give more accurate analysis of microstrip patch antenna characteristics. In this work, the authors extend the work on the use of the artificial neural network (ANN) technique taking into account of back propagation training algorithm with MLFFBP is stressed upon in place of conventional numerical techniques for the Microstrip antenna design. Then, with the ANN results, simulated results from the IE3D software are compared. The trained ANN is used to determine different important antenna characteristics for various structural input variables.

2. DESIGN AND DATA PRODUCTION

In this paper the L-slotted Microstrip antenna is designed to resonate at 5.5 GHz frequency with dielectric constant ε = 2.2, Substrate thickness h = 0.858 mm. Figure1a & Figure 1b. Shows the layout of a coaxial probe-fed L-slotted Microstrip Patch antenna.The width (W=21.56mm) and length (L=17.94mm) of antenna are calculated from given relationships.(1),(2),(3),(4) and (5) mention in [11].

$$W = \frac{c}{2f_o\sqrt{\frac{(\varepsilon_r + 1)}{2}}} \qquad (1)$$

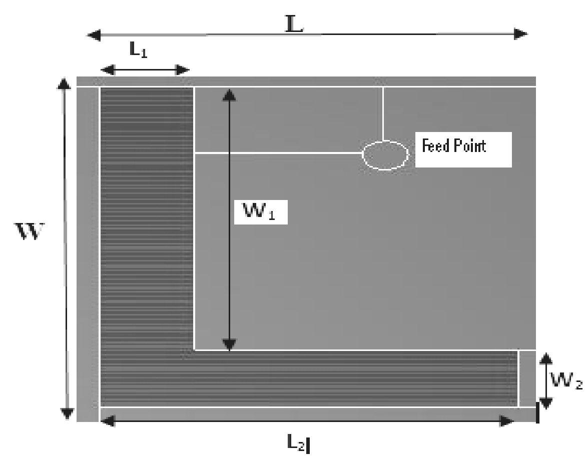

Figure.1a. L-slotted Microstrip Patch antenna.

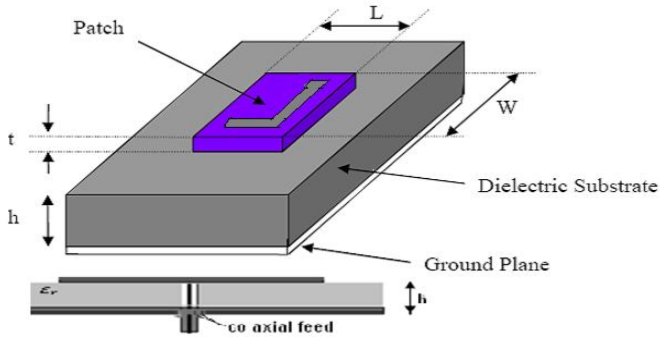

Figure.1b. .L-slotted Microstrip Patch antenna

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[1 + 12\frac{h}{W}\right]^{-\frac{1}{2}} \tag{2}$$

$$L_{eff} = \frac{c}{2f_o\sqrt{\varepsilon_{reff}}} \tag{3}$$

$$(\Delta L) = 0.412h\,\frac{\left(\varepsilon_{reff} + 0.3\right)\left(\frac{W}{h} + 0.264\right)}{\left(\varepsilon_{reff} - 0.258\right)\left(\frac{W}{h} + 0.8\right)} \tag{4}$$

$$L = L_{eff} - 2\Delta L \tag{5}$$

Hence for the design, the ground plane dimensions would be given as:

Lg = 6(h) + L = 6(0.858) + 17.94 = 25.01 mm
Wg = 6(h) + W = 6(0.858) + 21.56 = 28.73 mm.

For generating data, we simulated the frequency domain response of the antenna for various patch dimensions, using method of moments based simulation software IE3D^{TM}.For training and testing of the ANN, 120 data sets are generated by simulation using IE3D^{TM} simulation software. Figure.2 shows the return loss (S_{11}) = -21.58 dB for given physical dimensions for example antenna indicating that antenna is resonating at 7.22 GHz.

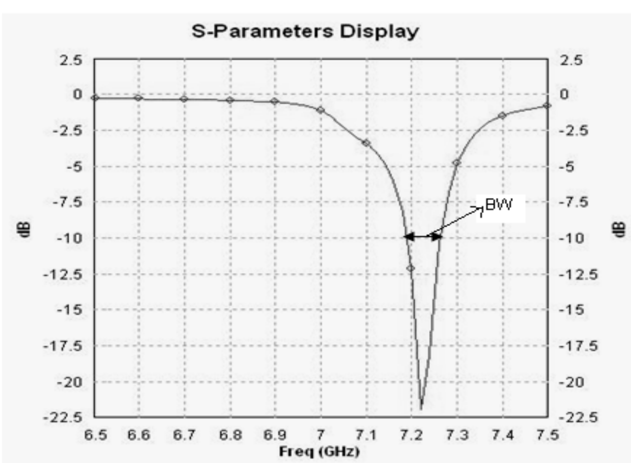

Figure.2 Return loss in dB vs. Resonating frequency of Microstrip antenna.

3. NETWORK ARCHITECHTURE

The artificial neural network model has been developed for Microstrip patch antenna shown in Figure.3

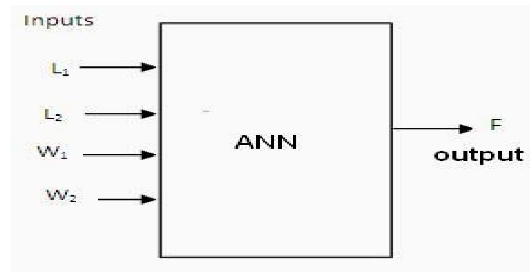

Figure.3 Analysis ANN Model [4].

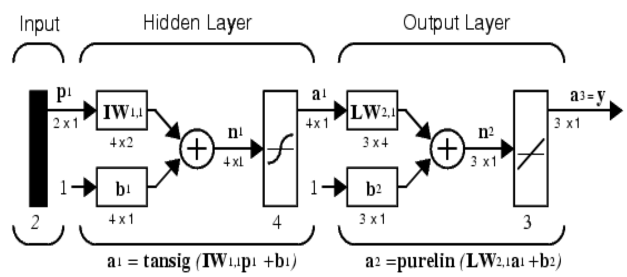

Figure.4 Multilayer feed forward neural network

The feed forward network has been utilized to calculate the outputs as resonance frequency (F) by putting the different values of dimensions L_1,L_2,W_1, and W_2 This defined as analysis ANN model. For the present work the different Multi-Layer Feed forward back Propagation neural networks [12, 13] and Radial basis function Artificial Neural Network models are used. These networks can be used as a general function approximator.It can be approximate any function with a finite number of discontinuous, arbitrary well given sufficient neurons in the hidden layer. The model is trained with 120 sets of input/output data, which are obtained by IE3D software based on MoM. The model is trained for different values of parameters (L_1,L_2,W_1, and W_2) to get desired outputs.

4. NEURAL NETWORK TRAINING

Back propagation neural network is usually based on the error back propagation to the multi-layer neural network,it is designed by D.E.Rumelhart and J.l.McCelland and its research term in 1986.Transfer function of BP neural network usually used Sigmoid function. It can be achieved between the input and output arbitrary nonlinear mapping. In this paper most common 6 back propagation algorithms and Radial basis function have used.

Multilayer Feed Forward Back Propagation (MLFFBP):

Multilayer Feed Forward Back Propagation is usually based on BP neurons in the multi-forward neural network structure. Typical back propagation structure shown in Figure4 .Back propagation [14] was created by Window-Hoff learning rule to multiple –layer networks and nonlinear differentiable transfer function. There are many variations of back propagation algorithm.The implementation of back propagation updates the network weights and biases .Six different variants of back propagation algorithms are used to improve the proposed design of network model.

Levenberg-Marquardt Optimization (LM)

The technique of direct search method though was appropriate for two parameter system, for higher parameter systems, the method would converge very slowly and usually did not converge on the values that would give a good fit. To overcome the problem we had to turn to techniques that would not only converge fast but also on the global minimum. For non – linear systems, the most popular technique used is Levenberg – Marquardt method for optimization. The L-M method falls in the broad class of Gradient method. It is an indirect method of optimization i.e. here the search for the minimum value was not carried out by determining several function values and then finding the minimum but considering that the function would have zero gradient at the extreme point. If F (x) is a function of n different independent variables denoted by vector x, then at the extreme point. we have $\nabla F(x) = 0$ where $\nabla F(x)$ is a partial derivative of F with respect to each variable. The negative gradient ($-\nabla F(x)$) gives the direction of steepest descent in an optimization process. Thus during optimization, the next best values are determined by moving in the direction of steepest descent that would take you to the minima. Search direction = s = $-\nabla F(x)$

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \lambda^{k} s^{k}$$

Where λ is the step size to be taken in the direction of descent and k is the k^{th} iteration during the search for the minima.

Quasi Newton algorithm (QN)

This is based on Newton's method but doesn't need calculation of second derivatives; an approximate Hessian Matrix is updated. Quasi-Newton methods require only the gradient (like steepest descent) of the objective to be computed at each iterate by successive measurements of the gradient. These methods build a quadratic model of the objective function which is sufficiently good that super linear convergence is achieved. Quasi-Newton methods are much faster than steepest descent methods. Since second derivatives (the Hessian) are not required, quasi-Newton methods are sometimes more efficient than Newton methods, especially when Hessian evaluation is slow/expensive.

Bayesian Regularization algorithm (BR)

A Bayesian network, Bayes network, belief network, Bayes(ian) model or probabilistic directed acyclic graphical model is a probabilistic graphical model (a type of statistical model) that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). `trainbr` is a network training function that updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that generalizes well. The process is called Bayesian regularization.

Conjugate gradient algorithms

The basic back-propagation algorithm adjusts the weights in the steepest descent direction (the most negative of the gradients). This is the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence.

Fletcher-Reeves Update (CGF):

All the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration.

$$\mathbf{p}_0 = -\mathbf{g}_0$$

A line search is then performed to determine the optimal distance to move along the current search direction:

$\mathbf{x}_{k+1} = \mathbf{x}_k \alpha_k \mathbf{p}_k$ .Then the next search direction is determined so that it is conjugate to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent direction with the previous search direction:

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad , \mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}$$

The various versions of the conjugate gradient algorithm are distinguished by [15]

Polak-Ribiére Update (CGP):

Another version of the conjugate gradient algorithm was proposed by Polak and Ribiére[16] . As with the Fletcher-Reeves algorithm, the search direction at each iteration is determined by

$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}$ For the Polak-Ribiére update, the constant β_k is computed by

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}}$$

Gradient descent with adaptive learning rate (GDA):

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. The function `traingdx` combines adaptive learning rate with momentum training. It is invoked in the same way as `traingda`, except that it has the momentum coefficient `mc` as an additional training parameter.

Radial Basis Function (RBF):

RBF network by the input layer, the hidden layer and a layer of the output before the three-tier network, to a single output of the neurons as an example, implied use of radial basis function as an incentive function, the RBF is generally Gaussian. The distance between each hidden layer neurons connected with the input of the weight w1i and the importation of vector xq, multiplied by the threshold b1i as their own input, as figure 5. Hidden layer of the i neural input than

$$k_i^q = \sqrt{\sum_j \left(w1_{ji} - x_j^q\right)^2} \times b1_i \quad (1)$$

Output:

$$r_i^q = exp\left(-k_i^{q^2}\right) = exp\left(\sqrt{\sum_j \left(w1_{ji} - x_j^q\right)^2} \times b1_i\right)$$

$$= exp\left(-\left(\left\| w1_i - x^q \right\| \times b1_i\right)\right)^2 \quad (2)$$

RBF threshold b1 can adjust the sensitivity function, however, in the actual work another parameter C (expansion constant) used more commonly. There are a number of established methods determine the relationship between C and b1,neural networks in Matlab Toolbox, the relation between C and b1" is

$$b1_i = 0.8326/C_j \quad (3)$$

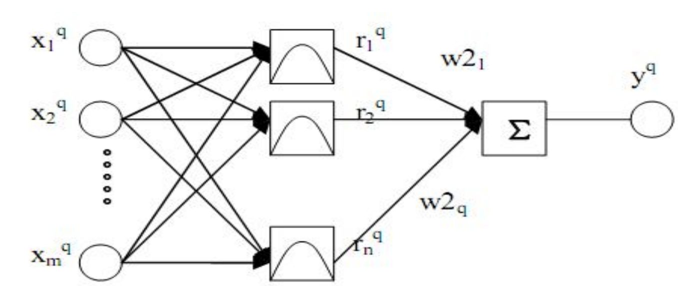

Figure.5 RBF neural network structure

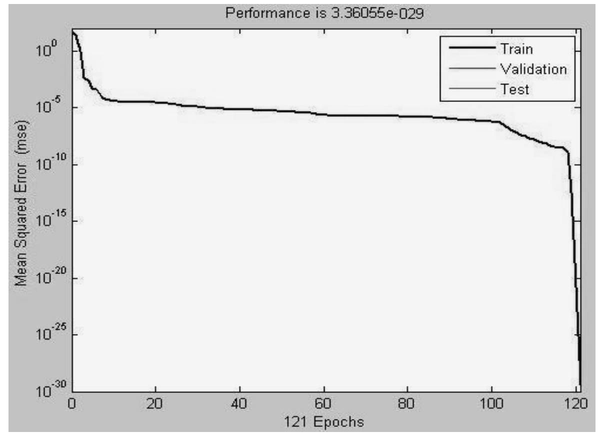

Figure. 6 Number of epochs vs. Mean square characteristics for Levenberg-Marquardt (LM) training algorithm

At this point, the output of hidden layer neurons:

$$g_j^q = exp\left[\frac{0.8326 \times \sqrt{\sum_j \left(w1_{ji} - x_j^q\right)^2}}{c_i}\right]$$

$$= exp\left[-0.8326^2 \times \left(\frac{\left\| w1_i - x^q \right\|}{c_i}\right)^2\right] \quad (4)$$

The weighted sum of the hidden layer neurons output as input data for the output:

$$y^q = \sum_{i=1}^{n} r_i \times w2_i \quad (5)$$

RBF network of training process in two steps: the first step to learning without teachers, determine weight w1 between input layer and hidden layer; the second step for the teachers learn to identify weight w2 between hidden layer and output layer. Before training, need to provide input vector X, target vector T and RBF expansion constant C.

5. RESULTS

From Table I it has been obtained that the Levenberg-Marquardt algorithm is suitable model to achieve more accuracy in MLFFBP.The network was trained with 120 train data and sample for test data. The 121 epochs used to minimize MSE level to 3.3605e-029 with tansig as a transfer function shown in Figure 6. In this paper different algorithms are used as Quasi Newton algorithm (QN), Bayesian Regularization algorithm (BR), Fletcher-Reeves Update (CGF), Polak-Ribiére Update (CGP), Gradient descent with adaptive learning rate (GDA) and Radial Basis Function (RBF).The mean square errors, absolute errors and epochs are used for various algorithms indicated in Table I. In this paper RBF ANN also used to implement the neural model shown in Figure 7.It is observed that the RBF (the spread value used of 0.01) gives more accurate results as well as low MSE value of 7.82392e-029 with only 49 epochs indicated in Table I.

TABLE I

COMPARISON OF DIFFERENT VARIANTS OF BACK PROPAGATION TRAINING ALGORITHM

Training algorithm	Epochs No.	MSE(Res. Frequency)		Absolute Error(Res. Frequency)	
		Train data	Test data	Train data	Test data
Levenberg-Marquardt (LM)	121	2.1849e-013	3.3605e-029	0.2738	0.6658
Quasi Newton algorithm (QN)	600	2.9489e-006	9.01022e-006	0.0443	0.3153
Bayesian Regularization (BR)	400	3.0647e-005	5.04505e-005	0.0107	0.0669
Fletcher-Reeves Update (CGF)	634	5.2460e-005	4.3763e-005	0.0629	0.5353
Polak-Ribiére Update (CGP)	423	4.8000e-005	3.9527e-005	0.0186	0.2751
Gradient descent adaptive learning rate (GDA)	520	1.99114e-005	2.7452e-005	0.3681	0.438
Radial Basis Function (RBF)	49	5.6798e-030	7.82392e-029	0.0027	0.0320

TABLE II

COMPARISON OF RESULTS OF IE3D SIMULATOR AND RBF ANN FOR CALCULATION RESONANCE FREQUENCY

S.No	Inputs				Outputs	
	L_1 (mm)	L_2 (mm)	W_1 (mm)	W_2 (mm)	Frequency (GHz) IE3D	Frequency (GHz) RBF ANN
1	2.1	9.9	10.9	2.2	7.348	7.455
2	2.1	10.2	10.7	2.3	7.321	7.4857
3	2	10.4	10.4	2	7.33	7.3959
4	2.4	10.8	11	2.1	7.275	7.2474
5	2.4	10.9	11.1	2.4	7.256	7.2177
6	2.3	10.4	11.2	2.3	7.293	7.3412
7	2.4	11.2	11.2	2.3	7.22	7.2229
8	2.2	9.2	9.8	2.4	7.437	7.4952
9	2.2	9.4	9.4	2	7.077	7.0712
10	2.3	9.4	10.4	2.4	7.412	7.641
11	2.3	9.8	10.3	2.3	7.376	7.2276
12	2	10.2	11.1	2.1	7.333	7.3717
13	2	9.7	9.7	2.2	7.412	7.1081

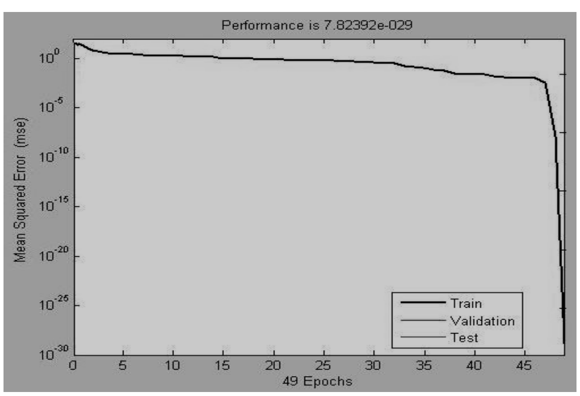

Figure. 7 Number of epochs vs. Mean square characteristics for RBF training algorithm

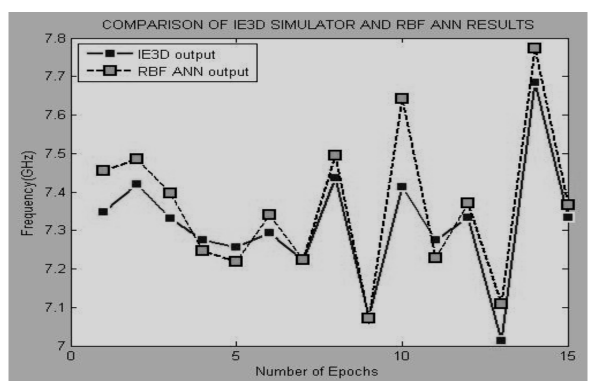

Figure. 8. Variation of results from IE3D and. RBF ANN results

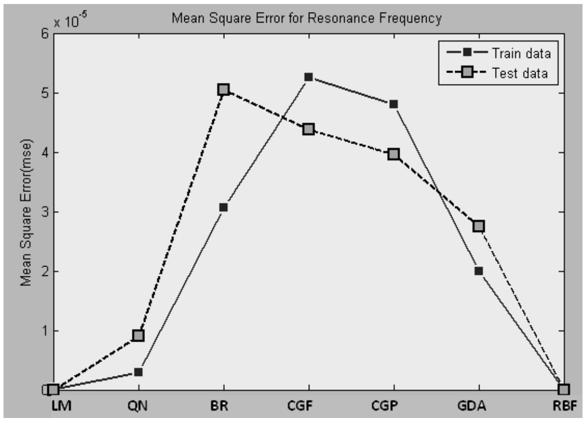

Figure. 9 Graph Showing variation of average mean square error for train data and test data

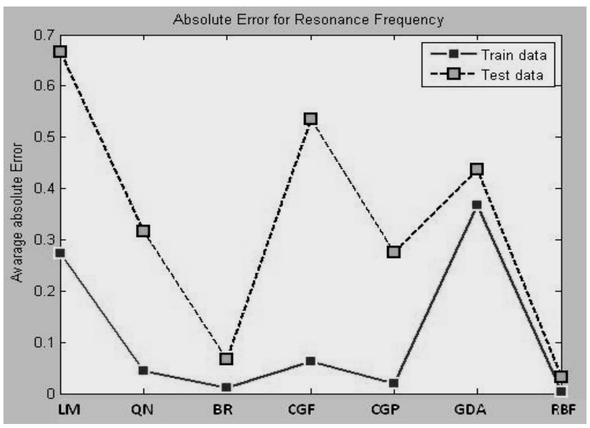

Figure. 10. Graph Showing variation of average mean square error for train data and test data.

In this paper resonance frequency obtained using IE3D simulator and RBF ANN are compared and found more accurate results indicated in Table II and the variation of results shown in Figure 8. Minimum MSE and average absolute errors of training and test data are shown in Figure 9 and Figure 10 respectively (Referred in Table I).

5. CONCLUSION

Multilayer Perceptron trained in the back propagation mode (using Levenberg-Marquardt algorithm) and Radial basis Function Network model are developed. The important characteristics namely, resonance frequency obtained with the present techniques is closer to the experimental results generated by simulating a large no. of L-slotted Microstrip patch antenna using IE3D software. The comprehensive comparison found that RBF network is better than MLFFBP network in prediction accuracy, training time and training speed. It proved that the RBF neural network is more efficient and accurate than MLPFFBP neural network. The developed neural network methodology can be extended for characterizing other different shapes of Microstrip antenna.

REFERENCES

[1] Balanis, C. A., Antenna Theory, John Wiley & Sons, Inc., 1997. J. Clerk Maxwell, **A Treatise on Electricity and Magnetism**, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] P. K. Singhal and L; Srivastava, "**On the investigations of a wide band proximity fed bow tie shaped microstrip antenna**", *Journal of Microwave and Optoelectronics*, Vol; 3, pp; 87-98, April 2004.

[3] Thakare V.V.,Singhal P.K.,"**Bandwidth Analysis by introducing slots in microstrip antenna design using ANN** " , *Progress In Electromagnetic Research M.*, Vol. 9,107-122,2009.

[4] Qi-Jan Zhang,Senior Member,IEEE,Kuldip C.Gupta, Fellow,IEEE, and Vijay K. Devabhaktuni, Student Member,IEEE"**Artificial Neural Networks for RF and Microwave Design-From Theory to Practice**", *IEEE Transaction on Microwave Theory and Techniques*, Vol. 51, No. 4,April 2003.

[5] Patnaik,A.;Anagnostou, D.;Christodoulou ,C.G.; Lyke, J.C., "**Neurocomputional Analysis of a Multiband Reconfigurable Planar Antenna**", *IEEE Transaction on Antennas and Propagation*, Vol. 53, No. 11, November 2005,p. 3453-3458.

[6] Patnaik,A.;Anagnostou, D.;Christodoulou ,C.G., "**Neural Networks in Antenna Engineering- Beyond Black –Box Modeling**", *IEEE/ACES International Conference on Wireless Communication and Applied Computional Electromagnetics*, April 2005,p. 598-601.

[7] Patnaik,A.;Mishra, R,K., "**ANN Techniques in Microwave Engineering**" *IEEE Microwave Magazine*, Vol. 1,No. 1, March 2000,p 55-60.

[8] Vegni,L.; Toscano, A.,"**Analysis of Microstrip Antennas Using Neural Networks**", *IEEE Transactions on Magnetics*, Vol. 33,No. 2,March 1997,p. 1414- 1419.

[9] Mishra, R,K.; Patnaik,A. **" Neurospectral Analysis of Coaxial Fed Rectangular Patch Antenna** ".*IEEE Antennas and Propagation Society International Symposium*, Vol. 2, July 2000,p. 1062-1065.

[10] Patnaik,A.;Anagnostou,D.E;Mishra,R.K.;Chistodoulou, C.G.; Lyke, J.C., "**Applications of Neural Networks in Wireless Communications**"*IEEE Antennas and Propagation Magazine*, Vol. 46,No. 3, June 2004,p. 130-137.

[11] D.M.Pozar,"**Microstrip Antennas**,"Jhon Wiley and Sons,Hoboken,1995,pp.79-81.doi:10.1109/5.119568.

[12] Haykin,S., **Neural Networks**, 2nd edition,PHI,2003.

[13] Hassoun, M.H., **Fundamentals of Artificial Neural Netorks** ,Chepter 8, New Delhi,Prentice Hall of India, 1999.

[14] S.Haykin,"**Neural Networks**,"2nd Edition,Prentice-Hall of india,Delhi,2000.

[15] G.P.Jaya Prakash and TRBstuff Representative,"**Use of Artificial Neural Networks In Geomechanical And Pavement System**",T*ransportation Rsearch Circular* ,Number E-co12,December 1999

[16] L.N.M.Tawfig and R.S.Naoum,"**On Training of Artificial Neural Networks**",*AL-Fath Journal*,No 23,2005

AUTHOR BIOGRAPHY

Mrs. Moumita Guha was born at kolkata,West Bengal, on December 24,1985.She received B.Tech degree in Instrumentation Engineering from West Bengal University of Technology,kolkata in 2008 and M.Tech degree in Instrumentation & Electronics Engineering from Jadavpur University,India, in 2012.Her current areas of interest are optimization of microstrip ,Radar syatem and metallic antennas.

Mr. Pritam Singha Roy was born at Berhampore,West Bengal,December 21,1981.He received M.Tech degree in Electronics and Control Engineering from Calcutta University,India, in 2009. He joined as Assistant professor at Dumkal Institute of Engineering and Technology,Murshidabd,in 2005.His current areas of interest are design, development,optimization of microstrip antennas.

Dr. samik Chakraborty was received Ph.D(Science) 2009 from Jadavpur University,India.He joined as an Assistant Professor at Indian Maritime University,Kolkata,India in 2006. He is published about thirty five research articles in referred journals and conference..He has worked in different research projects for National Space Research Organizations.