Progress In Electromagnetics Research M, Vol. 8, 119–129, 2009

NEURAL MODEL FOR CIRCULAR-SHAPED MICROSHIELD AND CONDUCTOR-BACKED COPLANAR WAVEGUIDE

P. Thiruvalar Selvan and S. Raghavan

Department of Electronics and Communication Engineering
National Institute of Technology
Tiruchirappalli 620015, Tamilnadu, India

Abstract—A Computer Aided Design (CAD) approach based on Artificial Neural Networks (ANN's) is successfully introduced to determine the characteristic parameters of Circular-shaped Microshield and Conductor-backed Coplanar Waveguide (CMCB-CPW). ANN's have been promising tools for many applications and recently ANN has been introduced to microwave modeling, simulation and optimization. The Multi Layered Perceptron (MLP) neural network used in this work were trained with Levenberg-Marquart (LM), Bayesian regularization (BR), Quasi-Newton (QN), Scaled Conjugate gradient (SCG), Conjugate gradient of Fletcher-Powell (CGF) and Conjugate Gradient backpropagation with Polak-Ribiere (CGP) learning algorithms. This has facilitated the usage of ANN models. The notable benefits are simplicity & accurate determination of the characteristic parameters of CMCBCPW's. The greatest advantage is lengthy formulas can be dispensed with.

1. INTRODUCTION

The principle of Coplanar Waveguides (CPWs) is that the location of ground planes is on the same substrate surface as the signal line. This simplifies the fabrication process by eliminating via holes. CPWs are often used in designing power dividers, balanced mixers, coupler and filters. The microshield lines, become a solution to technological problem in the design of coplanar line due to many advantages such as the ability to operate without the need for use of air bridges for ground equalization, reduced radiation loss and reduced electromagnetic

Corresponding author: P. Thiruvalar Selvan (tvs742002@yahoo.co.in).

interference. The other new type of coplanar structure is a Conductor-Backed CPW (CB-CPW) in which the lower ground plane is bent within the dielectric in a V-shape. This structure can reduce the current concentration at both edges of the strip conductor [1] and has the advantage of mechanical strength, heat sinking ability and lower characteristics impedance [2]. Addition to these advantages, CB-CPW also allows easy implementation of mixed coplanar/microstrip lines. The microshield microstrip lines have been studied using the conformal mapping technique [3] and the first analytic formulas for calculating quasi-static parameters of CB-CPWs were given by Lee [4]. The closed form design equation obtained by CMT is the simplest and most often used quasi-static method, consists of complete elliptic integrals which are difficult to calculate even with computer. To avoid this difficulties this paper proposed CAD-ANN based approach for the calculation of characteristic parameters of CMCBCPW using the approximate formulas with the use of one neural model. ANN recently gained attention as a fast and flexible tool to microwave modeling and design. Learning and generalization ability, fast real-time operation features made ANN's as popular. In microwave circuit components and micro strip antennas design applications, ANNs have more general functional forms and are usually better than the classical technique and provide simplicity in real-time operation.

2. DETERMINATION OF CHARACTERISTIC PARAMETERS OF CMCBCPW

The configuration taken for study is shown in Fig. 1 where the upper plane is deformed around inner conductor as circular shaped with the radius of b. The central conductor of width $2a$ is placed between the two planes, of spacing which are located on a substrate of thickness h, with relative permittivity ε_r. The distance between the strip conductor and short line is d. The slots are modeled as magnetic walls [5] and it is assumed that all metallic conductors are infinitely thin and perfectly conducting.

Based on this assumption, (to yield excellent results for practical line dimensions), the effective dielectric permittivity and characteristic impedance of the line by using conformal mapping are given by [6],

$$\varepsilon_{eff} = \frac{C(\varepsilon_r)}{C(1)}$$

$$Z_0 = \frac{\sqrt{\mu_0 \varepsilon_0}}{\sqrt{\varepsilon_{eff}}\,C(1)}$$

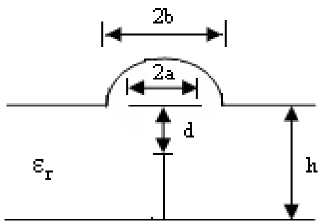


Figure 1. The Configuration of circular-shaped microshield and conductor-backed coplanar waveguide.

The total capacitance of the CMCBCPW,

$$\begin{aligned} C(\varepsilon_r) &= C_1(\varepsilon_r) + C_2(\varepsilon_r), \text{ and} \\ C_1(\varepsilon_r) &= 2\varepsilon_0\varepsilon_r \frac{K(k)}{K(k')} \text{ where } k = \frac{\tanh(\pi a/2h)}{\tan(\pi b/2h)} \\ C_2(\varepsilon_r) &= \frac{\pi\varepsilon_0\varepsilon_r}{\ln(2b/a)}, \end{aligned}$$

and $K(k)$ is the complete elliptic integral of first kind and $k'^2 = 1 - k^2$.

3. ARTIFICIAL NEURAL NETWORKS (ANNS)

ANN's are biologically inspired computer programs to simulate the way in which the human brain process information. It is a very powerful approach for building complex and nonlinear relationship between a set of input and output data [7, 8]. The power of computation comes from connection in a network. Each neuron has weighted inputs, simulation function, transfer function and output. The weighted sum of inputs constitutes the activation function of the neurons. The activation signal is passed through a transfer function which introduces non-linearity and produces the output. During training process, the inter-unit connections are optimized until the error in prediction is minimized [9]. Once the network is trained, new unseen input information is entered to the network to calculate the test output. There are many types of neural network for various applications available in the literature. The most commonly used and simplest network architecture called multilayered perceptron neural network (MLPNN) used in this work are feed-forward networks and universal approximators. A MLPNN consists of three layers: an input layer, an output layer and an intermediate or hidden layer. The neurons in the input layer only act as buffer for distributing the input signals to neuron in hidden layer. Each neuron in hidden layer sums up its input signal after weighting them and computes it outputs [10, 11]. Training a network consists of adjusting its weights using learning algorithms. The different training function used in this work has been explained briefly as follows:

3.1. Levenberg-Marquardt (LM) Algorithm

This is a least-square estimation method based on the maximum neighborhood idea and does not suffer from the problem of slow convergence. The LM method combines the best features of the Gauss-Newton technique and the steepest-descent method, but avoids many of their limitations [12, 13].

3.2. Bayesian Regularization (BR) Algorithm

This algorithm updates the weight and bias values according to their LM optimization and minimizes a linear combination of squared errors and weights. Here it is assumed that the weight and bias of the network are random variable. It is used to compute the Jacobean JX of performance with respect to weight and bias variable X. Each variable is adjusted according to LM.

$$dX = -\left[(JX) \times (JX) + \lambda \mathbf{I}\right]\left[(JX) \times E\right]^{-1}$$

where E is all errors and $\mathbf{I}$ is the identity matrix. Higher values continue to decrease the amount of memory needed and increase the training times. It also modifies the linear combination so that at the end of training the resulting network has good generalization qualities [14, 15].

3.3. Quasi-Newton (QN) Algorithm

This is based on Newton's method but doesn't require calculation of second derivatives [16]. At each iteration of the algorithm, the Hessian matrix ($\mathbf{A}_k$) update is computed as a function of the gradient. The line search function is used to locate the minimum. The first search direction is the negative of the gradient of performance. In succeeding iterations the search direction is computed according to the gradient. Newton's method often converges faster than conjugate gradient methods. The weight update for the Newton method is $w_{k+1} = w_k - g_k/\mathbf{A}_k$, $\mathbf{A}_k$ is the Hessian matrix of the performance index at the current value of the weights and biases.

3.4. Conjugate Gradient of Fletcher-Reeves (CGF)

In this algorithm, a search is performed along with conjugate directions, which produce generally faster convergence than steepest decent directions. Each variable is adjusted to minimize the performance along the search direction. The line search is used to locate the minimum point. The first search direction is the negative of the gradient. In succeeding iterations the search direction is computed from the new gradient and the previous search direction. Fletcher-Reeves version of conjugate gradient uses the norm square of previous gradient and the norm square of the current gradient to calculate the weights and biases. This algorithm updates weight and bias values according to the formulas proposed by Fletcher and Reeves [17]. The method of conjugate directions can be used to minimize a positive definite quadratic function in n steps.

3.5. Scaled Conjugate Gradient (SCG) Algorithm

This algorithm an optimization point of view, learning in a neural network is equivalent to minimizing a global error function, which is a multivariate function that depends on the weights in the network. Many of the training algorithms are based on the gradient descent algorithm. SCG belongs to the class of conjugate gradient methods, which show super linear convergence on the most problems. This was designed to avoid the time-consuming line approach. This algorithm is an implementation of avoiding the complicated line search procedure of conventional conjugate gradient algorithm [18].

3.6. Conjugate Gradient Back Propagation with Polak-Ribiere Algorithm (CGP)

This algorithm is a network training function that updates weight and bias values according to the conjugate gradient back propagation with Polak-Ribiere updates. It can train any network as long as its weight, net input, and transfer functions have derivative functions. It is used to calculate derivatives of performance with respect to the weight and bias variables. The line search function is used to locate the minimum point [19]. In succeeding iterations the search direction is computed from the new gradient. The search direction at each iteration is determined by updating the weight vector as: $w_{k+1} = w_k + \alpha p_k$, where $p_k = -g_k + \beta_k p_{k-1}$, $\beta_k = \frac{\Delta g_{k-1}^T g_k}{g_{k-1}^T g_{k-1}}$ and $\Delta g_{k-1}^T = g_k^T - g_{k-1}^T$.

4. DEVELOPMENT OF ANN MODEL FOR CMCBCPW

The MLP neural models have been successfully used to compute the effective permittivity (ε_{eff}) and characteristic impedances (Z_0) of CMCBCPW. The ranges of input data used are $0 \leq d/h \leq 1$; $0.1 \leq a/h \leq 0.9$; $0.5 \leq b/h \leq 1.5$. Training an ANN involves presenting those different sets (d/h, a/h, b/h and ε_r) sequentially and/or randomly and corresponding calculated values ε_{eff} and Z_0. Differences between the target and the actual outputs of the neural model (ε_{eff}-ANN and Z_0-ANN) are calculated through the network to adopt its weights. The adaptation is carried out after presenting each dataset (d/h, a/h, b/h and ε_r) until the calculation accuracy of the network is deemed satisfactory to the intension. The criterion includes RMS (root mean square) errors for all training set or the maximum allowable number of epochs to be reached. After many trail two hidden layered neural model (which provides high accuracy) have been selected as shown in Fig. 2. The suitable network configuration was $4 \times 12 \times 10 \times 2$. This means that the number of neurons were 4 for the input layer, 12 & 10 for the first & second hidden layers and 2 for the output layer. The hyperbolic tangent sigmoid activation functions were used for input and hidden layers and linear activation function was used in output layer.

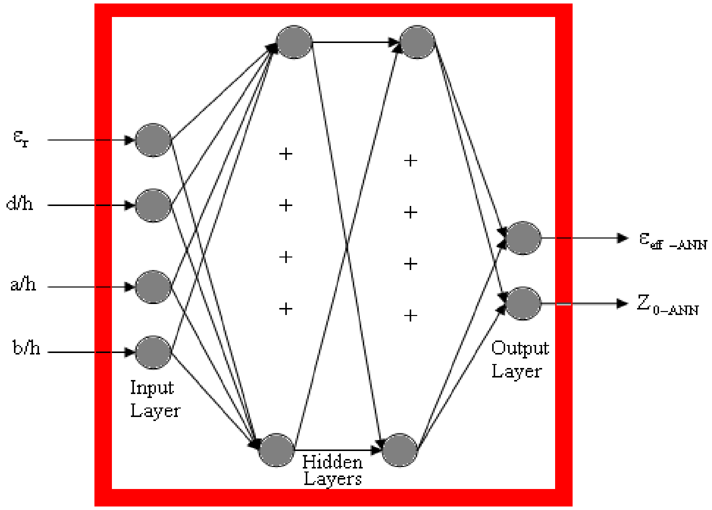


Figure 2. ANN structure for the determination of the characteristic parameters of CMCBCPW.

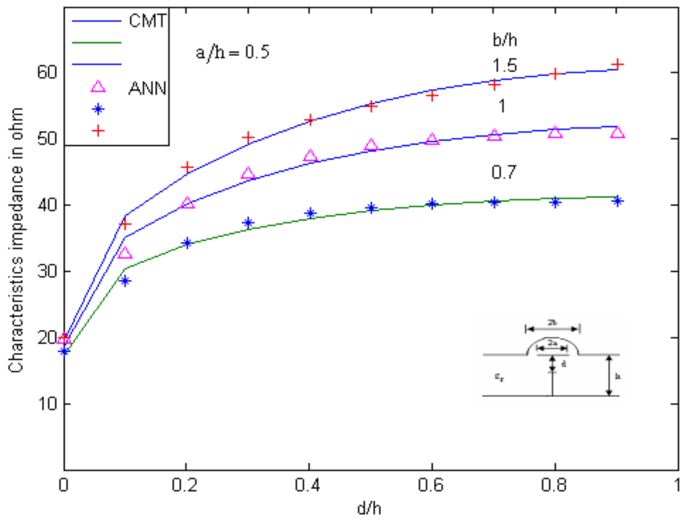


Figure 3. Comparison of the neural and CMT results for the characteristic impedance of CMCBCPW with $a/h = 0.5$, $\varepsilon_r = 2.55$.

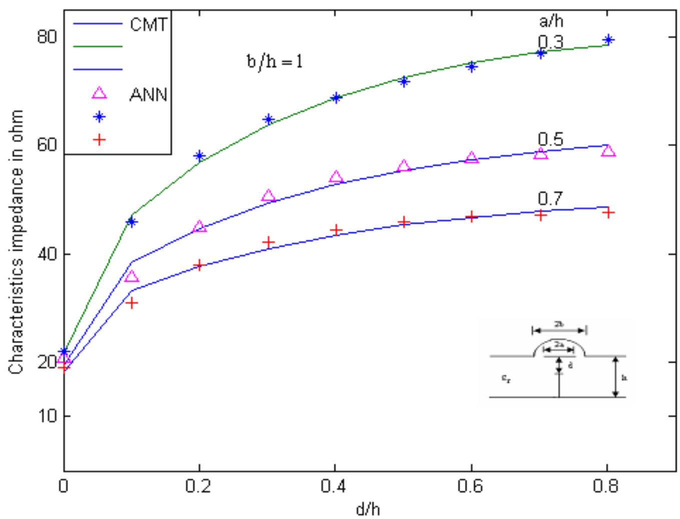
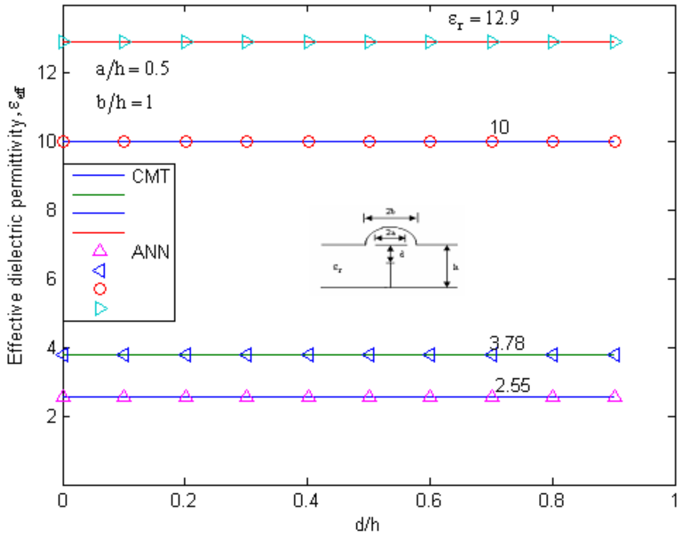
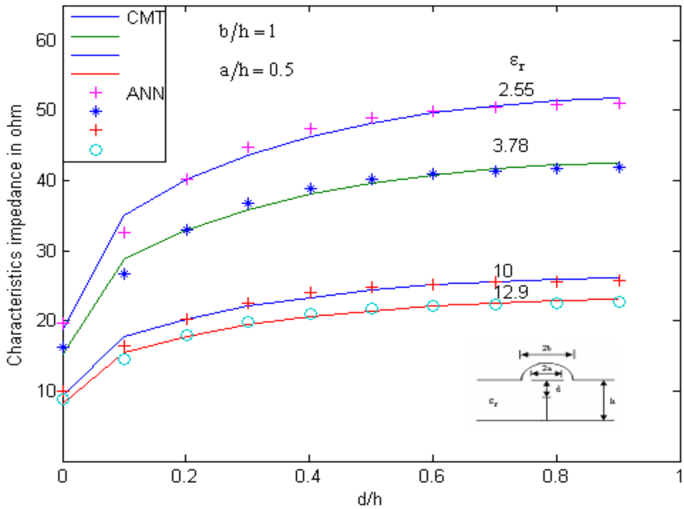


Figure 4. Comparison of the neural and CMT results for the characteristic impedance of CMCBCPW with $b/h = 1$, $\varepsilon_r = 2.55$.



(a)



(b)

Figure 5. Comparison of the neural and CMT results of characteristics parameters of CMCBCPW with $a/h = 0.5$, $b/h = 1$, $\varepsilon_r = 2.55$, 3.78, 10 and 12.9. (a) Effective dielectric permittivity and (b) characteristics impedance.

Table 1. Training and test RMS errors of the proposed neural model for ε_{eff} and Z_0.

Learning Algorithms	RMS errors in Training		RMS errors in Test	
	ε_{eff}	Z_0 in (Ω)	ε_{eff}	Z_0 in (Ω)
LM	1.4378e-017	5.9003e-004	4.3282e-016	0.0008
BR	1.2306e-022	2.5959e-004	2.4774e-018	9.4509e-004
QN	3.7288e-010	0.0012	1.5534e-009	0.1325
SCG	5.4963e-011	0.6202	7.2761e-008	0.9351
CGF	4.3580e-010	1.2007	1.0578e-005	1.5908
CGP	2.0224e-008	1.6227	3.2758e-006	1.8112

5. RESULTS & CONCLUSIONS

To obtain better performance, faster convergence and a simpler structure, the proposed ANN was trained with six different learning algorithms. The RMS errors obtained from the neural models for both the characteristic parameters are given Table 1. The comparison results of neural models and CMT were shown in Figs. 3, 4 and 5. The variation of characteristics parameters with respect to d/h for the values of $a/h = 0.5$, $\varepsilon_r = 2.55$, $b/h = 0.7$, 1 and 1.5 is shown in Fig. 3.

Figure 4 shows the variation characteristic impedance of CMCBCPW with respect to d/h for $b/h = 1$, $\varepsilon_r = 2.55$, $a/h = 0.7$, 0.5 and 0.3. Figs. 5(a) and 5(b) shows the characteristic parameters of CMCBCPW with respect to d/h for values of different dielectric substrates. The best ANN results were achieved from the models trained with BR and LM learning algorithms. From the figures and Table 1, it is clearly seen that the neural model results are in very good agreement with the CMT results.

So the neural model presented in this work is very successful for the determination of characteristic parameters of CMCBCPW. A distinct advantage of neural model computation is that after proper training, a neural network completely bypasses the repeated use of complex iterative processes for new cases presented to it.

The neural model presented in this paper achieves the determination of characteristic parameters in simple with high accuracy. It can be very useful for the development of fast CAD algorithms since it does not require complicated mathematics way.

REFERENCES

1. Yuan, N., C. Ruan, and W. Lin, "Analytical analyses of V, elliptic, and circular shaped microshield transmission lines," *IEEE Trans. Microwave Theory Tech.*, Vol. 42, 855–859, May 1994.
2. Simons, R. N., *Coplanar Waveguide Circuits, Components and Systems*, John Wiley & Sons, Inc., 2001.
3. Dib, N. I., W. P. Harokopus Jr., P. B. Katechi, C. C. Ling, and G. M. Rebeiz, "Study of a novel planar transmission line," *IEEE MTT-S Digest*, 623–626, 1991.
4. Lee, J.-W., I.-P. Hong, T.-H. Yoo, and H.-K. Park, "Quasi-static analysis of conductor backed coupled CPW," *IEEE Electronics Letters*, Vol. 34. No. 19, 1861–1862, Sep. 1998.
5. Gevorgian, S., L. J. Peter Linner, and E. L. Kollberg, "CAD models for shielded multilayered CPW," *IEEE Trans. Microwave Theory Tech.*, Vol. 43, 772–779, Apr. 1995.
6. Du, Z. and C. Ruan, "Analytical analysis of circular-shaped microshield and conductor-backed coplanar wave guide," *International Journal of Infrared and Millimeter Waves*, Vol. 18, No. 1, 165–171, 1997.
7. Yildiz, C. and M. Turkmen, "Quasi-static models based on artificial neural networks for calculating the characteristic parameters of multilayer cylindrical coplanar waveguide and strip line," *Progress In Electromagnetics Research B*, Vol. 3, 1–22, 2008.
8. Kaya, S., M. Turkmen, K. Guney, and C. Yildiz, "Neural models for the elliptic- and circular-shaped microshield lines," *Progress In Electromagnetics Research B*, Vol. 6, 169–181, 2008.
9. Zhang, Q. J. and K. C. Gupta, *Neural Networks for RF and Microwave Design*, Artech House, Boston, MA, 2000.
10. Haykin, S., *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Comp., 1994.
11. Yildiz, C., K. Guney, M. Turkmen, and S. Kaya, "Neural models for coplanar strip line synthesis," *Progress In Electromagnetics Research*, PIER 69, 127–144, 2007.
12. Fun, M.-H. and T. Martin Hagan, "Levenberg-marquardt training for modular networks," *Proceedings of the 1997 International Joint Conference on Neural Networks*, 468–473, 1996.
13. Levenberg, K., "A method for the solution of certain nonlinear problems in least squares," *Quarterly of Applied Mathematics*, Vol. 11, 431–441, 1963.
14. Mackay, D. J. C., "Bayesian interpolation," *Neural Computation*,

Vol. 3, No. 4, 415–447, 1992.

15. Foresee, F. D. and M. T. Hagan, "Gauss-Newton approximation to Bayesian regularization," *Proceedings of the 1997 International Joint Conference on Neural Networks*, 1930–1935, 1997.
16. Gill, P. E., *Practical Optimization*, Academic Press, New York, 1981.
17. Fletcher, R. and C. M. Reeves, "Function minimization by conjugate gradients," *Computer Journal*, Vol. 7, 149–154, 1964.
18. Moller, M. F., "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, Vol. 6, 525–533, 1993.
19. Dennis, E. and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, 1983.