# ARTIFICIAL NEURAL NETWORKS FOR THE CHARACTERISTIC IMPEDANCE CALCULATION OF CONDUCTOR-BACKED COPLANAR WAVEGUIDES

K. Güney, M. Erler, and S. Sagiroglu*

Department of Electronic Engineering, Faculty of Engineering, Erciyes University, 38039, Kayseri, Turkey, kguney@erciyes.edu.tr, erler@erciyes.edu.tr
*Department of Control and Computer Engineering, Faculty of Engineering, Erciyes University, 38039, Kayseri, Turkey, ss@posta.erciyes.edu.tr

**Abstract**
A new, simple method for calculating the characteristic impedance of conductor-backed coplanar waveguide, based on artificial neural networks, is presented. Three learning algorithms, the backpropagation, the delta-bar-delta, and the extended-delta-bar-delta, are used to train the networks. The method can be used for a wide range of substrate thicknesses and permittivities, and is useful for the computer-aided design (CAD) of coplanar waveguides. The calculated characteristic impedance results are in very good agreement with the results available in the literature.

## I. INTRODUCTION

Coplanar structures have received a lot of theoretical and experimental interest recently [1-3] because they have several advantages over conventional microstrips for use in monolithic or hybrid integrated circuit applications at microwave frequencies, including easy parallel and series insertion of both active and passive components and high circuit density. Several rigorous methods [1-3] are available to determine accurately the characteristic impedance of conductor-backed coplanar waveguide (CPW), as this is one of the most popular coplanar structures. Exact mathematical formulations in rigorous methods involve extensive numerical procedures, resulting in round-off errors, and may also need final experimental adjustments to the theoretical results. They are also time consuming and not easily included in a CAD system. For these reasons, in this work a new simple method based on artificial neural networks (ANNs) for calculating the characteristic impedance of conductor-backed CPW has been presented. Conductor backing of the substrate improves both the mechanical strength and the power-handling capability [1-4]. Ability and adaptability to learn, generalizability, smaller information requirement, fast real-time operation, and ease of implementation features have made artificial neural networks popular in the last few years [5-19]. Because of these fascinating features, artificial neural networks in this article are used to calculate the characteristic impedance of conductor-backed CPW.

In previous works [12-19], we also successfully introduced the artificial neural networks to model a robot sensor, and to compute the various parameters of the triangular, rectangular and circular microstrip antennas. In the most of these works, only the backpropagation algorithm was used to train the neural model. However, in this study three learning algorithms, the backpropagation (BP), the delta-bar-delta (DBD), and the extended-delta-bar-delta (EDBD), are used to train the networks. The reason for using three different learning algorithms is to speed up the training time and to improve the performance of neural models.

In the following sections, the coplanar waveguides, the artificial neural networks, and the application of the networks to the calculation of the characteristic impedance of conductor-backed CPW are explained.

## II. COPLANAR WAVEGUIDES

The CPW proposed by Wen [20] consists of two slots each of width W printed on a dielectric substrate, as shown in Fig. 1(a). The thickness and relative dielectric constant of the substrate are denoted by h and $\varepsilon_r$, respectively. The spacing between the slots is S. The conventional CPW presented by Wen [20] can not be used as such because of the requirement of infinitely thick substrate. For practical applications substrate thickness has to be finite as in Fig. 1(b). It also very tempting to introduce a conductor backing to improve the mechanical strength of the transmission line so that thin substrates can be used. The certain way of computing the characteristic impedance of conductor-backed CPW with finite dielectric thickness involves the quasi-static and fullwave analyses [1-3]. Wen [20] presented a quasi-static analysis of the coplanar lines using conformal mapping, with the assumption that the dielectric substrate is thick enough to be considered infinite. Conformal transformation has also been applied to take into account the effects of the finite thickness of the dielectric substrate, finite size of the ground planes, upper shield, ground plane below the substrate as in microstrip line, structural asymmetry, and multilayer configuration. A fullwave analysis of coplanar lines, which provides information regarding frequency dependence of phase velocity and characteristic impedance, has been carried out by a number of authors [1-3]. The techniques employed include Galerkin's method in the spectral domain, variational methods, integral equation, relaxation method, method of lines, mode-matching technique, and finite-difference time-domain technique [1-3]. These methods require high performance large-scale computer resources and a very large number of computations.

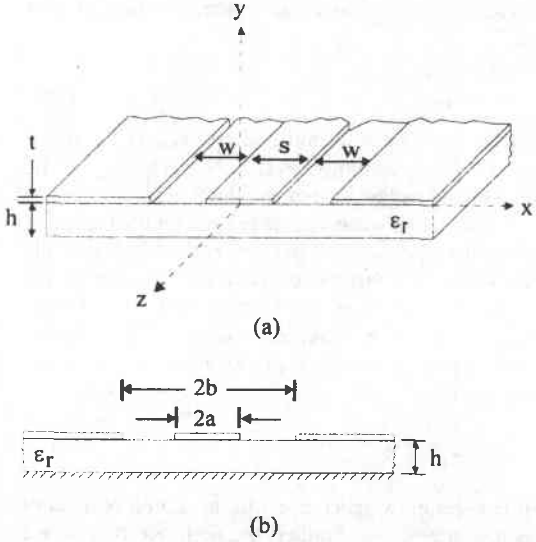



**Fig.1** Geometry of (a) CPW and (b) conductor-backed CPW.

It is clear from all of the methods [1-3] proposed in the literature that only four parameters, $\varepsilon_r$, h, a, and b, are needed to describe the characteristic impedance of conductor-backed CPW. In this work, the characteristic impedance is calculated by using a new model based on artificial neural networks. Only four parameters, $\varepsilon_r$, h, a, and b, are used in calculating the characteristic impedance.

## III. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks have many structures and architectures [5-8]. Multilayered perceptrons (MLPs) [5-8] are the simplest and therefore most commonly used neural network architectures. They have been adapted for the calculation of the characteristic impedance of conductor-backed CPW. MLPs can be trained using many different learning algorithms [5-11]. In this work, MLPs are trained with the three supervised learning algorithms, the BP, the DBD, and the EDBD [5, 7-10]. As shown in Figure 2, an MLP consists of three layers: an input layer, an output layer and an intermediate or hidden layer. Processing elements (PEs) or neurons (indicated in Figure 2 with the circle) in the input layer only act as buffers for distributing the input signals $x_i$ to PEs in the hidden layer. Each PE j in the hidden layer sums up its input signals $x_i$ after weighting them with the strengths of the respective connections $w_{ji}$ from the input layer and computes its output $y_j$ as a function *f* of the sum, viz.

$$y_i = f(\sum w_{ji} x_i) \qquad (1)$$

*f* can be a simple threshold function, a sigmoidal or hyperbolic tangent function. The output of PEs in the output layer is computed similarly.

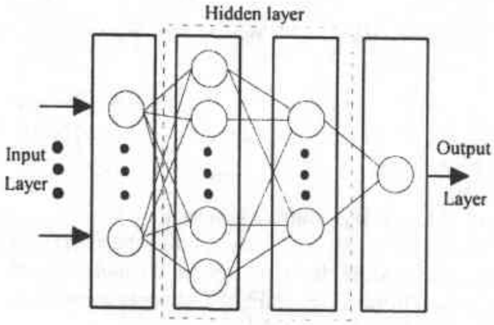



**Fig.2** General form of backpropagation multilayered perceptron.

Training a network consists of adjusting weights of the network using the different learning algorithms [5,7-11]. A learning algorithm gives the change $\Delta w_{ji}(k)$ in the weight of a connection between PEs i and j. In the following section, three learning algorithms used in this study have been explained briefly.

**Backpropagation Algorithm**
The algorithm [5] is the most commonly adopted MLP training algorithm. It is a gradient descent algorithm and gives the change $\Delta w_{ji}(k)$ in the weight of a connection between PEs *i* and *j* as follows

$$\Delta w_{ji}(k) = \eta \delta_j x_i + \alpha \Delta w_{ji}(k-1) \qquad (2)$$

where $\eta$ is a parameter called the learning coefficient, $\alpha$ is the momentum coefficient, and $\delta_j$ is a factor depending on whether PE *j* is an output PE or a hidden PE.

**Delta-Bar-Delta Algorithm**
The DBD algorithm is a heuristic approach to improve the speed of convergence of the connection weights in MLPs [8,9]. The connection weight is updated by

$$w(k+1) = w(k) + \alpha(k)\delta(k) \qquad (3)$$

where $\alpha(k)$ is learning coefficient and assigned to each connection, $\delta(k)$ is the gradient component of the weight change. $\delta(k)$ is employed to implement the heuristic for incrementing and decrementing the learning coefficients for each connection [9]. The weighted average $\bar{\delta}(k)$ is formed as

$$\bar{\delta}(k) = (1-\theta)\delta(k) + \theta\bar{\delta}(k-1) \quad (4)$$

where $\theta$ is the convex weighting factor. The connection learning coefficient change is given as

$$\Delta\alpha(k) = \begin{cases} \kappa, & \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi\alpha(k), & \bar{\delta}(k-1)\delta(k) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $\kappa$ is the constant learning coefficient increment factor, and $\varphi$ is the constant learning coefficient decrement factor.

**Extended Delta-Bar-Delta Algorithm**

This algorithm [8,10] is an extension of the DBD and based on decreasing the training time for multilayered perceptrons. In this algorithm, the changes in weights are calculated as

$$\Delta w(k+1) = \alpha(k)\delta(k) + \mu(k)\Delta w(k) \quad (6)$$

and the weights are then found as

$$w(k+1) = w(k) + \Delta w(k+1) \quad (7)$$

In eq.(6), $\alpha(k)$ and $\mu(k)$ are the learning and momentum coefficients, respectively.

The learning coefficient change is given as

$$\Delta\alpha(k) = \begin{cases} \kappa_\alpha \exp(-\gamma_\alpha|\bar{\delta}(k)|), & \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_\alpha\alpha(k), & \bar{\delta}(k-1)\delta(k) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $\kappa_\alpha$ is the constant learning coefficient scale factor, *exp* is the exponential function, $\varphi_\alpha$ is the constant learning coefficient decrement factor, and $\gamma_\alpha$ is the constant learning coefficient exponential factor. The momentum coefficient change is also written as

$$\Delta\mu(k) = \begin{cases} \kappa_\mu \exp(-\gamma_\mu|\bar{\delta}(k)|), & \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_\mu\mu(k), & \bar{\delta}(k-1)\delta(k) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\kappa_\mu$ is the constant momentum coefficient scale factor, $\varphi_\mu$ is the constant momentum coefficient decrement factor, and $\gamma_\mu$ is the constant momentum coefficient exponential factor.

As can be seen from eqns.(8)-(9), the learning and the momentum coefficients have separate constants controlling their increase and decrease. $\delta(k)$ is used whether an increase or decrease is appropriate. The adjustment for decrease is identical in form to that for the DBD explained above. Therefore, the increases in the both coefficients were modified to be exponentially decreasing functions of the magnitude of the weighted gradient components $|\bar{\delta}(k)|$. Thus, greater increases will be applied in areas of small slope or curvature than in areas of high curvature. This is a partial solution to the jump problem. In order to take a step further to prevent wild jumps and oscillations in the weight space, ceilings are placed on the individual connection learning and momentum coefficients. For this,

$$\alpha(k) \le \alpha_{max}$$
$$\mu(k) \le \mu_{max} \quad (10)$$

must be for all connections, where $\alpha_{max}$ is the upper bound on the learning coefficient, and $\mu_{max}$ is the upper bound on the momentum coefficient.

Finally, after each epoch presentation of training tuples, the accumulated error is evaluated [8]. If the error $E(k)$ is less than the previous minimum error, the weights are saved as the current best. A recovery tolerance parameter $\lambda$ controls this phase. Specifically, if the current error exceeds the minimum previous error such that

$$E(k) > E_{min}\,\lambda \quad (11)$$

All connection weights revert to the stored best set of weights in memory. Further, the both coefficients are decreased to begin the recovery.

## IV. APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO THE CALCULATION OF THE CHARACTERISTIC IMPEDANCE

The proposed method involves training MLPs to calculate the characteristic impedance of conductor-backed CPW when the values of $\varepsilon_r$, h, a, and b, are given. Figure 3 shows the neural model used in neural computation of the characteristic impedance. As explained before the three learning algorithms, the BP, DBD, and the EDBD, are used to train MLPs. In the three MLPs, the input and output layers have the linear transfer function and the hidden layer has the tangent hyperbolic function. Training the MLPs by three learning algorithms to compute the characteristic impedance involves presenting them sequentially with different ($\varepsilon_r$, h, a, b) sets and corresponding target values. Differences between the target output and the actual output of the MLPs are trained through the three learning algorithms to adapt their weights. The adaptation is carried out after the presentation of each set ($\varepsilon_r$, h, a, b) until the calculation accuracy of the network is deemed satisfactory according to the root-mean-square (rms) error between the target output and the actual output for all the training sets that fall below 0.01 or the maximum allowable number of epochs is reached to 150,000.

The training and test data sets used in this paper have been obtained from the previous works [1-4]. A set of random values distributed uniformly between −0.1 and +0.1 was used to initialize the weights of the three networks. However, the input data tuples were scaled between -1.0 and +1.0 and the output data tuples were also scaled between −0.8 and +0.8 before training. The most suitable network configuration found was 10 PEs for the hidden layer. Both sequential and random procedures were used in training.

The parameters of the networks are: *for BP*, the learning coefficients were set to 0.3 for hidden layer and 0.15 for the output layer, and the momentum coefficient was also set to 0.3; *for DBD*, $\kappa$=0.01, $\varphi$=0.5, $\theta$=0.7, $\alpha$=0.2, the momentum coefficient was fixed to 0.4; *for EDBD*, $\kappa_\alpha$=0.095, $\kappa_\mu$=0.01, $\gamma_\mu$=0.0, $\gamma_\alpha$=0.0, $\varphi_\mu$=0.01, $\varphi_\alpha$=0.1, $\theta$=0.7, $\lambda$=1.5.

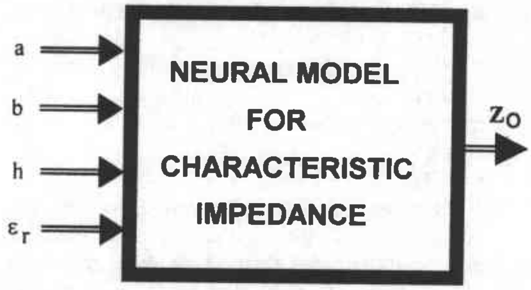
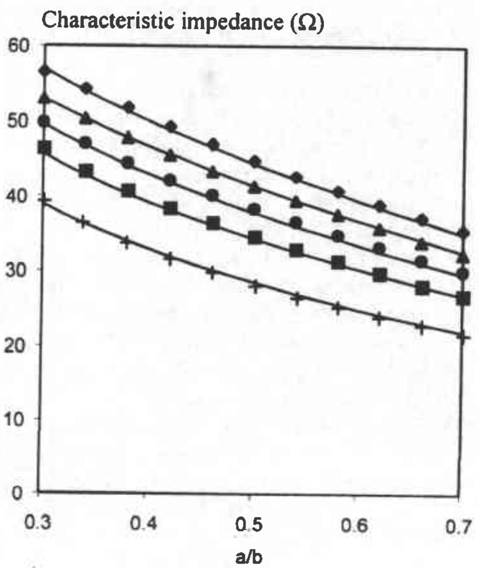


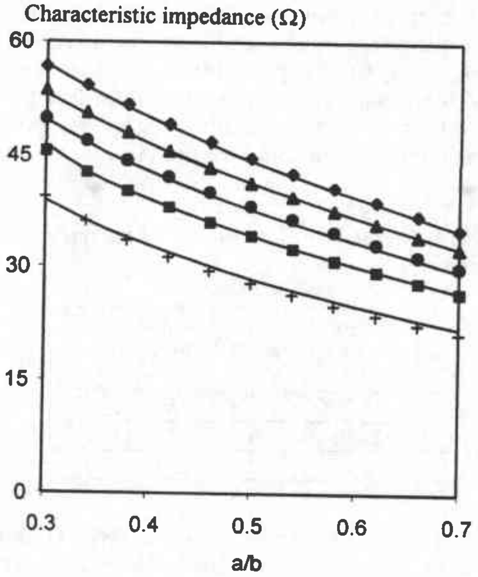

**Fig.3** Neural model for characteristic impedance calculation.

## V. RESULTS AND CONCLUSIONS

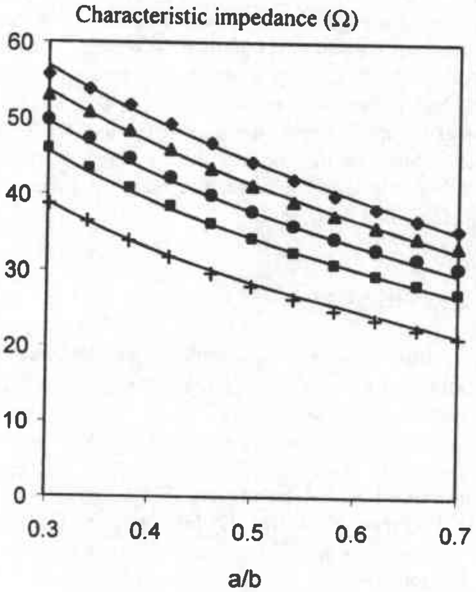
In order to demonstrate the computational effort of the neural models, the test results of ANNs for $\varepsilon_r$=13 with different values of h, a, and b which are not used in the training process are compared with the results of well-known reliable method [2] in Fig.4. The test results




(a)




(b)




(c)

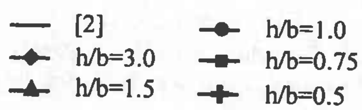



**Fig.4** Characteristic impedance results of conductor-backed CPW ($\varepsilon_r$=13) for (a) EDBD, (b) DBD and (c) BP learning algorithms.

illustrate that the performance of the proposed method is quite robust and precise. As can be seen from Fig.4, there is an excellent agreement with the data from the method [2]. This excellent agreement supports the validity of the ANNs.

For three different learning algorithms, the number of iteration when the rms error is set to 0.01 is given in Table I. When the performances of three neural models are compared with each other, the best result was obtained from the EDBD. Among the neural models, the worst performance was obtained from the BP.

**Table I.** The number of iteration when the rms error is set to 0.01

| Learning algorithms used in training | The number of iteration |
|---|---|
| EBDB | 86,426 |
| DBD | 118,384 |
| BP | 127,428 |

Since the ANNs presented in this work have high accuracy and require no complicated mathematical functions, they can be very useful for the development of fast CAD algorithms. The advantages of the ANNs given here are simplicity and accuracy. The proposed ANNs do not require the complicated Green's function methods and integral transformation techniques. The ANNs only require four parameters: $\varepsilon_r$, h, a, and b. For engineering applications, the simple models are very usable. Thus the ANNs trained by three learning algorithms can also be used for many engineering applications and purposes.

## VI. REFERENCES

[1] T. Itoh, *Numerical Techniques for Microwave and Millimeter-wave Passive Structures*, Wiley, New York, 1989

[2] R. Goyal (Ed.), *Monolithic Microwave Integrated Circuits: Technology and Design*, Norwood, MA:Artech House, 1989.

[3] K.C.Gupta, R. Garg, I. Bahl and P. Bhartia, *Microstrip Lines and Slotlines*, Artech House, Boston, 1996.

[4] G. Ghione and C. Naldi, "Parameters of coplanar waveguides with lower ground plane," *Electron. Lett.*, vol. 19, pp. 734-735, 1983.

[5] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing,* vol.1, Cambridge, MA: MIT Press, 1986.

[6] A. Maren, C. Harston, and R. Pap, *Handbook of Neural Computing Applications,* London: Academic Press, ISBN 0-12-471260-6, 1990.

[7] S. Haykin, *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing Company, ISBN 0-02-352761-7, 1994.

[8] *Neural Computing, A Technology Handbook for Professional II/PLUS and NeuralWorks Explorer*, Pittsburgh: NeuralWare, Inc., Technical Publications Group, 1996.

[9] R. A. Jacobs, "Increased rate of convergence through learning rate adaptation," *Neural Networks*, vol.1, pp. 295-307, 1988.

[10] A. A. Minai and R. D. Williams, "Acceleration of backpropagation through learning rate and momentum adaptation," *Int. Joint Conf. on Neural Networks*, vol.1, pp. 676-679, Jan. 1990.

[11] S. E. Fahlman, "An Emprical Study of Learning Speed in Backpropagation Networks," *Technical Report CMU-CS-88-162*, Carnegie Mellon University, June 1988.

[12] D. T. Pham and S. Sagiroglu, "Three methods of training multi-layer perceptrons to model a robot sensor," *Robotica*, vol.13, pp. 531-538, 1996.

[13] D. T. Pham and S. Sagiroglu, "Synergistic neural models of a robot sensor for part orientation detection," *IMechE, Proc. Instn. Mech. Engrs.*, vol.210, pp. 69-76, 1996.

[14] S. Sagiroglu and K. Güney, "Calculation of resonant frequency for an equilateral triangular microstrip antenna using artificial neural networks," *Microwave Opt. Technol. Lett.*, vol.14, no.2, pp. 89-93, 1997.

[15] S. Sagiroglu, K. Güney, and M. Erler, "Neural computation of mutual coupling coefficient of electrically thin and thick rectangular microstrip antennas," *Proc. of International Conference on Neural Network and Brain (NN&B'98)*, Beijing, China, pp. 223-226, Oct. 27-30,1998.

[16] S. Sagiroglu, K. Güney, and M. Erler, "Resonant frequency calculation for circular microstrip antennas using artificial neural networks," *Int. J. of RF Microwave and Millimeter-Wave Computer-Aided Engineering*. vol.8, pp. 270-277, 1998.

[17] K. Güney, M. Erler, and S. Sagiroglu, "Neural computation of mutual coupling coefficient between two rectangular microstrip antennas with various substrate thicknesses," *Proc. of PIERS'98*, Nantes, France, pp. 57, July 13-17, 1998.

[18] S. Sagiroglu, K. Güney, and M. Erler, "Calculation of bandwidth for electrically thin and thick rectangular microstrip antennas with the use of multilayered perceptrons," *Int. J. of RF and Microwave Computer-Aided Engineering*, vol.9, pp. 277-286, 1999.

[19] D. Karaboga, K. Güney, S. Sagiroglu, and M. Erler, "Neural computation of resonant frequency of electrically thin and thick rectangular microstrip antennas," *IEE Proceedings-Microwaves, Antennas and Propagation*, Pt.H., 1999, accepted for publication.

[20] C.P. Wen, "Coplanar waveguide: A surface strip transmission line suitable for non-reciprocal gyromagnetic device application", *IEEE Trans. Microwave Theory and Tech.*, MTT-17, pp. 1087-1090, 1969.