Erc. Üniv. Fen Bil. Derg. 15, 1-2(1999)32-48

CALCULATION OF DIRECTIVITY FOR THE E-PLANE AND H-PLANE SECTORAL HORN ANTENNAS WITH THE USE OF ARTIFICIAL NEURAL NETWORKS

Kerim GÜNEY, Mehmet ERLER, and Şeref SAĞIROĞLU*

Department of Electronic Engineering, Faculty of Engineering, Erciyes University, 38039, Kayseri, Turkey, kguney@erciyes.edu.tr

*Department of Computer Engineering, Faculty of Engineering, Erciyes University, 38039, Kayseri, Turkey.

ABSTRACT

A method for calculating the directivity of the E-plane and H-plane sectoral horn antennas, based on the artificial neural networks, is presented. The extended-delta-bar-delta algorithm is used to train the networks. The directivity results obtained by using this method are in very good agreement with the results available in the literature.

E-DÜZLEM VE H-DÜZLEM SEKTÖREL HUNİ ANTENLERİN YAPAY SİNİR AĞLARI İLE YÖNELTİCİLİK HESABI

ÖZET

Yapay sinir ağlarına dayanan bir metod, E-düzlem ve H-düzlem sektörel huni antenlerin yönelticiliğini hesaplamak için sunulmuştur. Ağları eğitmek için, genişletilmiş delta-bar-delta algoritması kullanılmıştır. Bu metod kullanılarak elde edilen yönelticilik sonuçları, literatürdeki mevcut sonuçlarla uyumluluk içindedir.

1. INTRODUCTION

The electromagnetic horn is a widely used antenna at microwave frequencies due to its simplicity in construction, ease of excitation, versatility, large gain, and preferred overall performance [1-2]. Its most common uses include feed elements of reflector antennas and calibration standards. The horn antenna consists of a waveguide which has its walls flared into a wider opening at one end. Both circular and rectangular waveguides are commonly used. Rectangular horn antennas can take on several forms. The simplest form is that of an open-ended waveguide. This antenna is not actually a horn, but simply a rectangular waveguide with one end left open, from which the internal electromagnetic fields propagate. Horn antennas, then, can be viewed as open-ended waveguides with a flared wall section added to the open end. These antennas with walls flared only in the direction parallel to the E-field and H-field are known as the E-plane and H-plane sectoral horn antennas, respectively. A horn antenna flared in both the E-plane and H-plane is known as a pyramidal horn antenna.

The one way of computing the directivity of the horn antennas involves the complicated sine and cosine Fresnel integrals [1-2]. In this work, a simple method based on artificial neural networks (ANNs) for calculating the directivity of the E-plane and H-plane sectoral horn antennas has been presented. ANNs [3-4] are developed from neurophysiology by morphologically and computationally mimicking human brains. Although the precise operation details of artificial neural networks are quite different from human brains, they are similar in three aspects: they consist of a very large number of processing elements (the neurons), each neuron connects to a large number of other neurons, and the functionality of networks is determined by modifying the strengths of connections during a learning phase.

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 [3-21]. Because of these fascinating features, artificial neural networks in this article are used to calculate the directivity of the E-plane and H-plane sectoral horn antennas. First, the antenna parameters related to the directivity are determined, then the directivity depending on these parameters are calculated by using the neural model. The directivity results calculated by using this model are in very good agreement with the results reported elsewhere. The extended-delta-bar-delta (EDBD) algorithm [11] is used to train the network.

In previous works [14-21], 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.

2. DIRECTIVITY OF SECTORAL HORN ANTENNAS

Consider the E-plane and H-plane sectoral horn antennas, as shown in Figure 1. The directivity of the E-plane sectoral horn antenna can be expressed as [1-2]

$$D_E = \frac{64a\rho_1}{\pi\lambda b_1}\left[C^2\left(\frac{b_1}{\sqrt{2\lambda\rho_1}}\right)+S^2\left(\frac{b_1}{\sqrt{2\lambda\rho_1}}\right)\right] \tag{1}$$

where a and b_1 are the aperture dimensions, ρ_1 is the horn length, λ is the wavelength, and C(x) and S(x) are the cosine and sine Fresnel integrals.

The directivity of the H-plane sectoral horn antenna is given by [1-2]

$$D_H = \frac{4\pi b \rho_2}{a_1 \lambda} \left\{ [C(u) - C(v)]^2 + [S(u) - S(v)]^2 \right\} \tag{2}$$

with

$$u = \frac{1}{\sqrt{2}} \left(\frac{\sqrt{\lambda \rho_2}}{a_1} + \frac{a_1}{\sqrt{\lambda \rho_2}} \right) \tag{2a}$$

$$v = \frac{1}{\sqrt{2}} \left(\frac{\sqrt{\lambda \rho_2}}{a_1} - \frac{a_1}{\sqrt{\lambda \rho_2}} \right) \tag{2b}$$

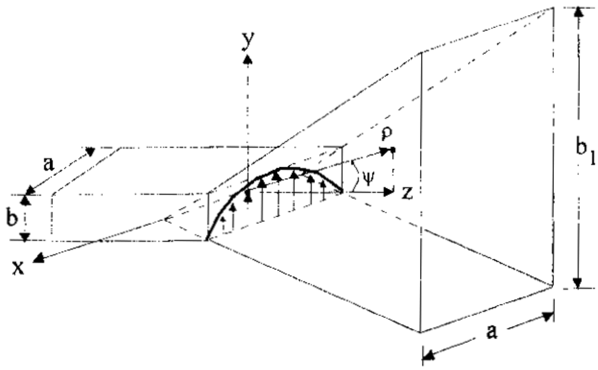
where a_1 and b are the aperture dimensions, and ρ_2 is the horn length.

The directivity of a pyramidal horn can be written as [1-2]

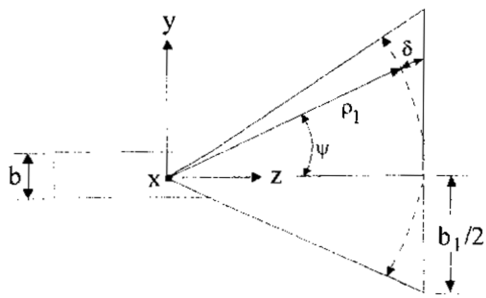
$$D_p = \frac{\pi \lambda^2}{32ab} D_E D_H \tag{3}$$

It is clear from eqns. (1)-(3) that the aperture dimensions, horn length and wavelength are needed to describe the directivity.

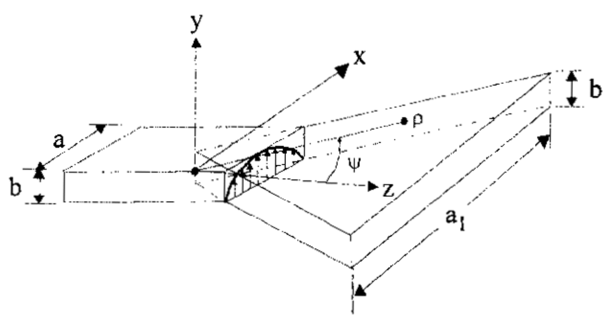
In the following sections, the artificial neural networks and the EDBD used in training the networks are described briefly and the application of the networks to the calculation of the directivity of the sectoral horn antennas is then explained.



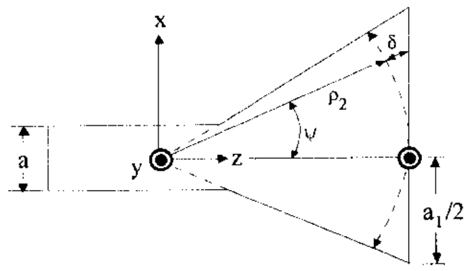
(a) E-plane horn



(b) E-plane view



(c) H-plane horn



(d) H-plane view

Figure 1. E-plane and H-plane sectoral horn antennas.

3. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks have many structures and architectures [4-5]. Multilayered perceptrons (MLPs) [3-5] are the simplest and therefore most commonly used neural network architectures. They have been adapted for the calculation of the directivity of the sectoral horn antennas. MLPs can be trained with the use of many different learning algorithms [3-8,11]. In this work, the EDBD algorithm [11] has been used for training MLP. 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.,

$$y_i = f\left(\sum w_{ji} x_i\right) \tag{4}$$

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

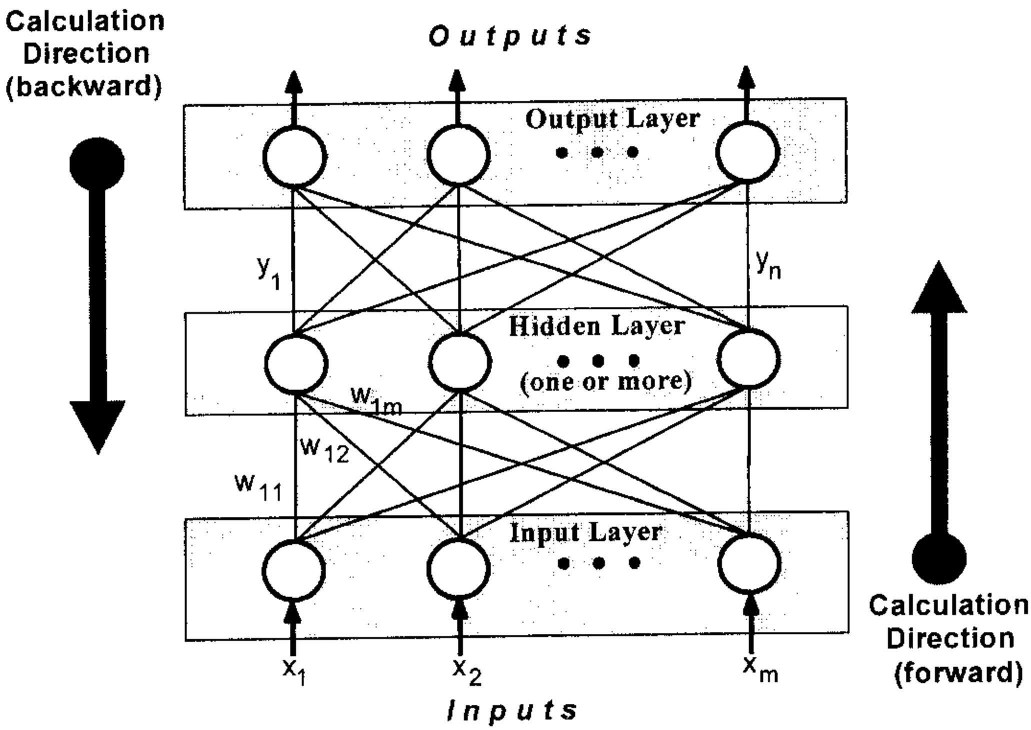


Figure 2. General form of neural networks.

Training a network consists of adjusting weights of the network using the learning algorithms. 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, the EDBD learning algorithm used in this study has been explained briefly.

Extended Delta-Bar-Delta Algorithm

This algorithm is an extension of the delta-bar-delta algorithm [8] and based on decreasing the training time for multilayered perceptrons. The use of the momentum heuristics and avoiding the cause of the wild jumps in the weights are the features of the algorithm developed by Minai and Williams [11]. The EDBD algorithm includes a little-used 'error recovery' feature which calculates the global error of the current epoch during training [11]. If the error measured during the current epoch is greater than the error of the previous epoch, then the network's weights revert back to the last set of weights (the weights which produced the lower error).

However, a patience factor has been included [6] into the error recovery feature, which may produce the better performance of the networks through the use of this feature. Instead of testing the error upon every epoch, as was performed previously, the error is now tested upon n-th epoch, where n equals the patience factor. In this algorithm, the changes in weights are calculated as

$$\Delta w(k+1)= \alpha(k)\delta(k) +\mu(k)\, \Delta w(k) \tag{5}$$

and the weights are then found as

$$w(k+1)= w(k) + \Delta w(k+1) \tag{6}$$

In eqn. (5), $\delta(k)$ is the gradient component of the weight change, and $\alpha(k)$ and $\mu(k)$ are the learning and momentum coefficients, respectively. $\delta(k)$ is employed to implement the heuristic for incrementing and decrementing the learning coefficients for each connection [8]. The weighted average $\bar{\delta}(k)$ is formed as

$$\bar{\delta}(k)=(1-\theta)\,\delta(k)+\theta\,\delta(k-1) \tag{7}$$

where θ is the convex weighting factor.

The learning coefficient change is given as

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

where κ_α is the constant learning coefficient scale factor, *exp* is the exponential function, φ_α is the constant learning coefficient decrement factor, and γ_α is the constant learning coefficient exponential factor. The momentum coefficient change is also written as

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

where κ_μ is the constant momentum coefficient scale factor, φ_μ is the constant momentum coefficient decrement factor, and γ_μ 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 delta-bar-delta algorithm. 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 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,

$$\begin{aligned} \alpha(k) &\leq \alpha_{max} \\ \mu(k) &\leq \mu_{max} \end{aligned} \qquad (10)$$

must be for all connections, where α_{max} is the upper bound on the learning coefficient, and μ_{max} is the upper bound on the momentum coefficient.

Finally, after each epoch presentation of training tuples, the accumulated error is evaluated [6]. If the error $E(k)$ is less than the previous minimum error, the weights are saved as the current best. A recovery tolerance parameter λ controls this phase. Specifically, if the current error exceeds the minimum previous error such that

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

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

4. APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO THE CALCULATION OF THE DIRECTIVITY

The proposed method involves training a neural network to calculate the directivity (D) when the values of the aperture dimensions, horn length, and wavelength are given. In this work, the normalized directivity (relative to the constant aperture dimension a for the E-plane sectoral horn antenna and b for the H-plane sectoral horn antenna) is calculated. A neural model used in →

calculating the D for the E-plane sectoral horn antenna is shown in Figure 3. In the MLP, the input and output layers have the linear transfer function and the hidden layers have the tangent hyperbolic function. Training an MLP with the use of the EDBD algorithm to compute the D involves presenting them sequentially with different (the aperture dimensions, horn length, and wavelength) sets and corresponding target values. Differences between the target output and the actual output of the MLP are trained through the EDBD algorithm to adapt their weights. The adaptation is carried out after the presentation of each set (the aperture dimensions, horn length, and wavelength) until the calculation accuracy of the network is deemed satisfactory according to some criterion (for example, when the root-mean-square (rms) error between the target output and the actual output for all the training set falls below a given threshold) or the maximum allowable number of epochs is reached.

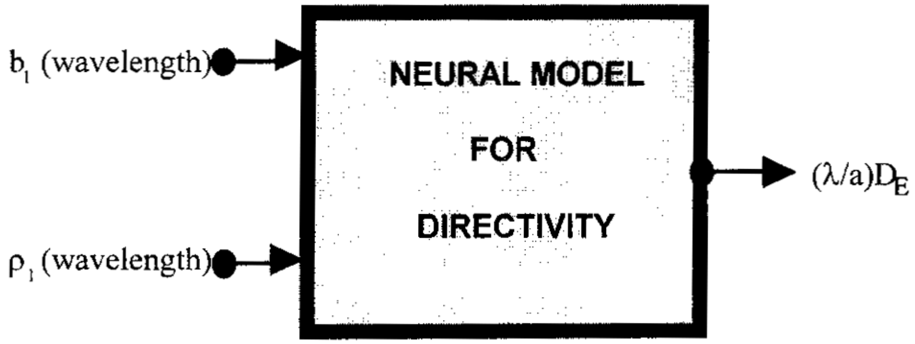


Figure 3. Neural model for directivity.

The training and test data sets used in this paper have been obtained from the previous works [1-2]. The 350 data sets for training and 30 data sets for testing were used for both the E-plane and H-plane sectoral horn antennas.

A set of random values distributed uniformly between -0.1 and +0.1 was used to initialize the weights of the 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 number of epoch was 300,000 for training.

After several trials, it was found that two layers network achieved, as indicated in [9-10,12], the task with high accuracy. The most suitable network configuration found was 10 PEs and 5 PEs for the first and the second hidden layers, respectively. The seed number was fixed to 257. Both sequential and random procedures were used in training. The parameters of the networks for EDBD are: κ_α=0.095, κ_μ=0.01, γ_μ=0.0, γ_α=0.0, φ_μ=0.01, φ_α=0.1, θ=0.7, λ=1.5.

5. RESULTS AND CONCLUSIONS

In order to demonstrate the computational effort of the neural model, the test results of ANN for ρ_1= 30λ and ρ_2=30λ which are not used in training process are compared with the results of well-known reliable method [1-2] in Figures 4-5. The test results illustrate that the performance of the proposed method is quite robust and precise. As can be seen from Figures 4-5, there is excellent agreement with the data from the method [1-2]. This excellent agreement supports the validity of ANN.

When the directivities of the E-plane and H-plane sectoral horn antennas are known, the directivity of a pyramidal horn antenna can be computed by -?

using eqn. (3). Thus, the neural directivity results for the E-plane and H-plane sectoral horn antennas can be used for calculating the directivity of a pyramidal horn antenna.

In this work, the different learning algorithms such as the backpropagation, the delta-bar-delta, and the quick propagation were also used to train the networks. However, the best results was obtained from the EDBD. For this reason, only the results of the EDBD were given in this paper. In our previous work [20], the best bandwidth results of microstrip antennas were also obtained by using the EDBD.

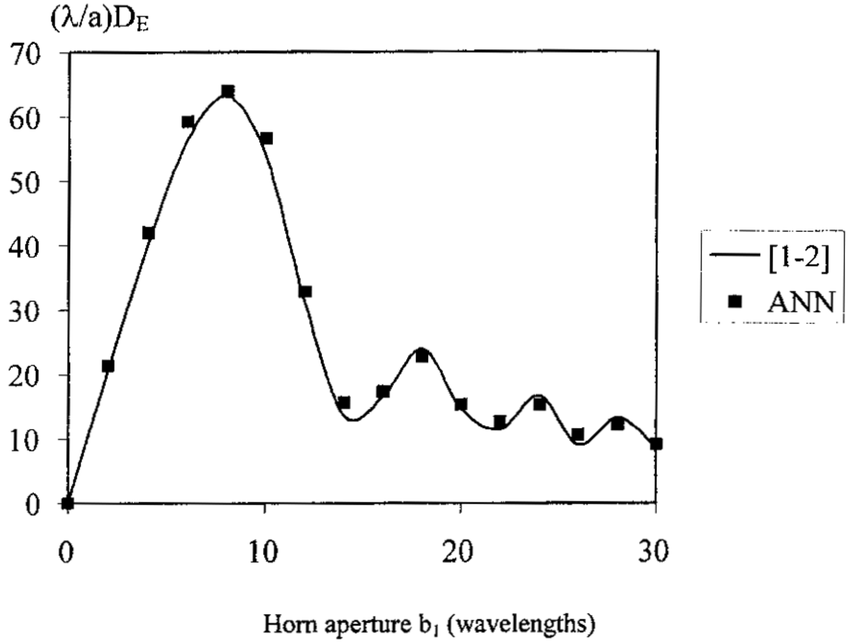


Figure 4. Normalized directivity of E-plane sectoral horn as a function of aperture size and for $\rho_1=30\lambda$.

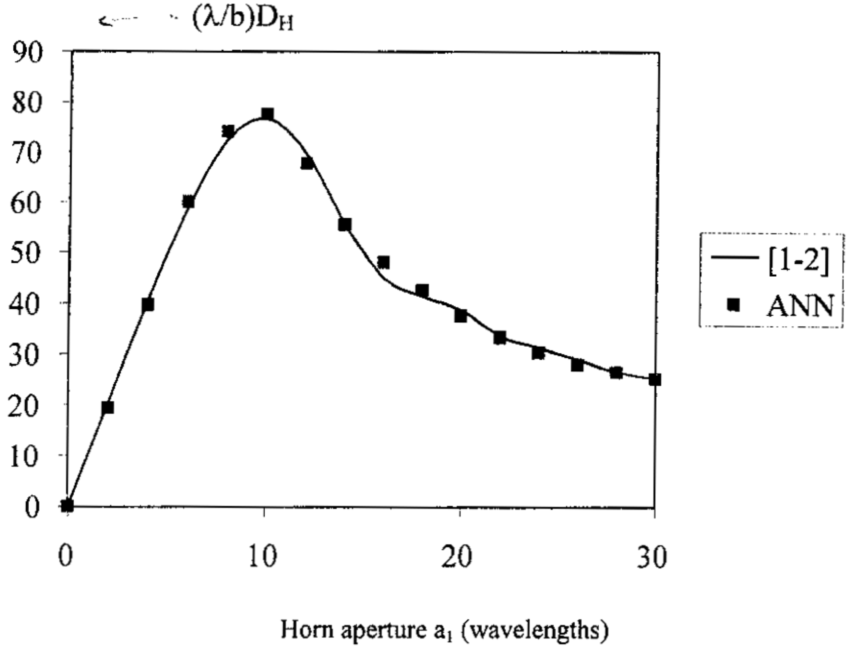


Figure 5. Normalized directivity of H-plane sectoral horn as a function of aperture size and for $\rho_2=30\lambda$.

A distinct advantage of neural computation is that, after proper training, a neural network completely bypasses the repeated use of complex iterative processes for new cases presented to it. These proposed neural model does not require the computation of cosine and sine Fresnel integrals. The model only requires the aperture dimensions, horn length, and wavelength. For engineering applications, the simple models are very usable. Thus the neural model given in this work can also be used for many engineering applications and purposes.

6.REFERENCES

[1] C.A. Balanis, "Antenna Theory: Analysis and Design", John Wiley & Sons, New York, 1982.

[2] E.A. Wolff, "Antenna Analysis", Artech House, Norwood, 1988.

[3] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing", Vol.1, The MIT Press, Cambridge, 1986.

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

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

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

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

[8] R. A. Jacobs, "Increased Rate of Convergence Through Learning Rate Adaptation", Neural Networks, 1, 295-307, (1988).

[9] G. Mirchandani and W. Cao, "On Hidden Nodes for Neural Nets", IEEE Transactions on Circuits and Systems, 36, no.5, 661-664, (1989).

[10] D. L. Chester, "Why Two Hidden Layers Are Better Than One", Proc. Int. Joint Conf. on Neural Networks, Washington, DC, 1, 265-268, (1990).

[11] A. A. Minai and R. D. Williams, "Acceleration of Backpropagation Through Learning Rate and Momentum Adaptation", Int. Joint Conf. on Neural Networks, 1, 676-679, (Jan. 1990).

[12] W. Y. Huang and Y. F. Huang, "Bounds on the Number of Hidden Neurons in Multilayered Perceptron", IEEE Trans. on Neural Networks, 1, no.4, 47-55, (1991).

[13] P. Burrascano, S. Fiori, and M. Mongiardo, "A Review of Artificial Neural Networks Applications in Microwave Computer-Aided Design", Int. J. of RF and Microwave Computer-Aided Engineering, 9, 158-174, (1999).

[14] D. T. Pham and S. Sagiroglu, "Three Methods of Training Multi-Layer Perceptrons to Model a Robot Sensor", Robotica, 13, 531-538, (1996).

[15] D. T. Pham and S. Sagiroglu, "Synergistic Neural Models of a Robot Sensor for Part Orientation Detection", IMechE, Proc. Instn. Mech. Engrs., 210, 69-76, (1996).

[16] S. Sagiroglu and K. Güney, "Calculation of Resonant Frequency for an Equilateral Triangular Microstrip Antenna Using Artificial Neural Networks" Microwave Opt. Technol. Lett., 14, no.2, 89-93, (1997).

[17] 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, 223-226, Oct. 27-30, (1998).

[18] 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, 8, 270-277, (1998).

[19] 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, 57, (July 13-17, 1998).

[20] 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, 9, 277-286, (1999).

[21] 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., 146, 155-159, (1999).