Osmangazi Üniversitesi Müh. Mim. Fak. Dergisi C. XII, S. 2, 1999
Eng. & Arch. Fac., Osmangazi University, Vol. XII, No. 2, 1999

NEURODESIGN OF LOG-PERIODIC DIPOLE ARRAYS

Kerim GÜNEY[1], Mehmet ERLER[1], Şeref SAĞIROĞLU[2]

ABSTRACT: *The design of log-periodic dipole arrays, based on artificial neural networks, is presented. The extended-delta-bar-delta algorithm is used to train the networks. The design results obtained by using the neural models are in very good agreement with the results available in the literature.*

KEY WORDS: *log-periodic dipole arrays, artificial neural networks*

LOG-PERİYODİK DİPOL DİZİLERİN NÖRAL TASARIMI

ÖZET: *Yapay sinir ağlarına dayanan bir metot, log-periyodik dizilerinin tasarımı için sunulmuştur. Ağları eğitmek için, genişletilmiş delta-bar-delta algoritması kullanılmıştır. Nöral modeller kullanılarak elde edilen tasarım sonuçları, literatürdeki mevcut sonuçlarla uyumluluk içindedir.*

ANAHTAR KELİMELER: *Log-periyodik dipol dizileri, yapay sinir ağları*

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

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

I. INTRODUCTION

The log-periodic dipole arrays have been well-known in the literature [1-3] for its broad bandwidth capability. This wide band operation is achieved by different groups of elements radiating at different frequencies. The spacing between the elements is proportional to their length and the system is fed using a transmission line. As the frequency ratio varies, the elements that are at or near resonance, couple energy from the transmission line. The resulting radiation pattern is directional and has a broadly constant radiation characteristics over the full operating frequency range. The general configuration of a log-periodic dipole array is described in terms of the design parameters: the spacing factor σ, the scale factor τ, and the angle α.

The most important log-periodic dipole array design procedure that meets given specifications has been presented by Carrel [3]. As a design aid, the directivity contour curves as a function of τ for various values of σ and the graphs of the relative characteristic impedance of a feeder line as a function of relative characteristic impedance of dipole element were also given in [3]. However, these design curves are not usable for computer aided design (CAD) and practical usage of engineering applications. For this reason, in this work, the optimum values of σ and τ for a given directivity and the relative characteristic impedance of a feeder line as a function of relative characteristic impedance of a dipole element for various values of relative mean spacings σ' are obtained by using a simple method based on artificial neural networks (ANNs). After obtaining these parameters, the design of log-periodic dipole arrays is very easy.

ANNs [4-5] 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 [4-22]. Because of these fascinating features, artificial neural networks in this article are used to design the log-periodic dipole arrays. The extended-delta-bar-delta (EDBD) algorithm [12] is used to train the networks.

The design results obtained by using the neural models are in very good agreement with the results reported elsewhere [1-3, 23]. The model is simple, and very useful to antenna engineers for accurately design of log-periodic dipole arrays. In previous works [15-22], 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 following section, the design procedure of a log-periodic dipole arrays is given. This design procedure follows the design procedure presented in [1].

II. LOG-PERIODIC DIPOLE ARRAYS

The log periodic dipole arrays are tapered linear arrays of dipole elements of varying lengths that operate over a wide frequency range. The geometry of a log periodic dipole array and associated connections is shown in Fig. 1. The lengths (l_n's), spacings (R_n's), diameters (d_n's), and gap spacings at dipole centers (s_n's) are related as follows [1]

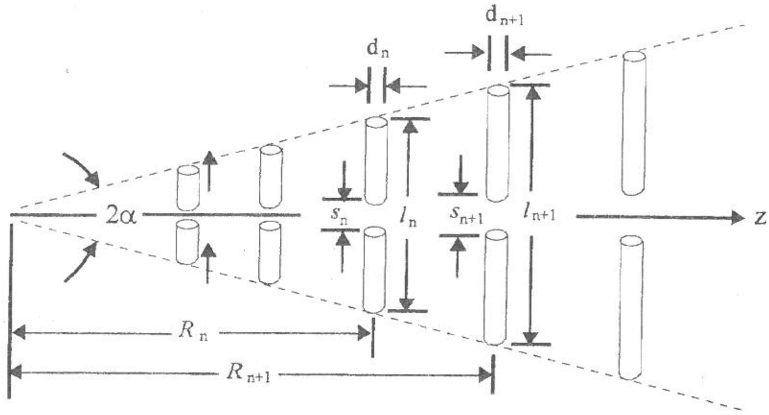
$$\frac{1}{\tau}=\frac{l_2}{l_1}=\frac{l_{n+1}}{l_n}=\frac{R_2}{R_1}=\frac{R_{n+1}}{R_n}=\frac{d_2}{d_1}=\frac{d_{n+1}}{d_n}=\frac{s_2}{s_1}=\frac{s_{n+1}}{s_n} \tag{1}$$

where τ is the scale factor. The spacing factor σ is given by

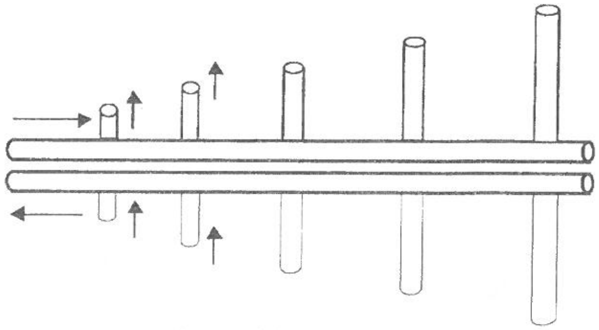
$$\sigma=\frac{R_{n+1}-R_n}{2l_{n+1}} \tag{2}$$

The design parameters σ, τ, and α are related as follows

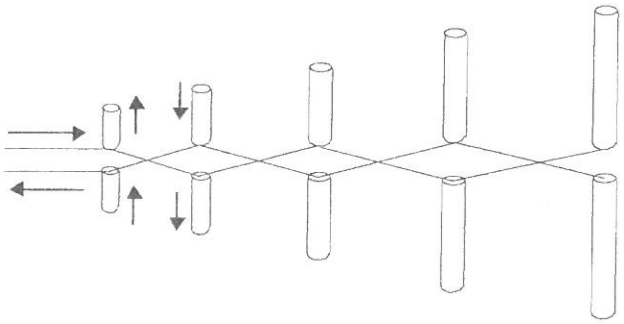
$$\alpha=tan^{-1}\left[\frac{1-\tau}{4\sigma}\right] \tag{3}$$



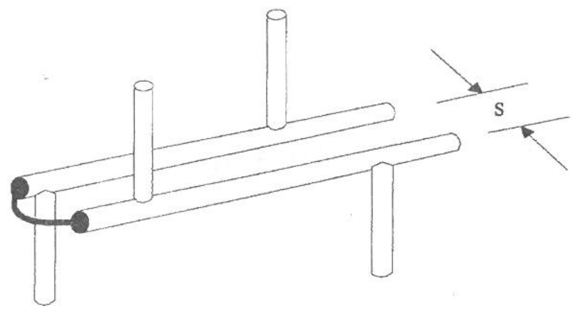
(a)Dipole array



(b) Straight connection



(c) Crisscross connection



(d) Coaxial connection

Figure 1. Log-periodic dipole array and associated connections.

The designed bandwidth is

$$B_s = B\left[1.1 + 7.7(1-\tau)^2 \cot\alpha\right] \tag{4}$$

where B is the desired bandwidth. The total length of the array L, from the shortest l_{min} to the longest l_{max} element, is determined by

$$L = \frac{\lambda_{max}}{4}\left(1 - \frac{1}{B_s}\right)\cot\alpha \tag{5}$$

with

$$\lambda_{max} = 2l_{max} = \frac{c}{f_{min}} \tag{6}$$

where f_{min} is the lower frequency limit of desired operation, and c is the velocity of electromagnetic waves in free space. The number of elements can be written as

$$N = 1 + \frac{\ln(B_s)}{\ln\frac{1}{\tau}} \tag{7}$$

The average characteristic impedance of the elements is given by

$$Z_a = 120\left[ln\left(\frac{l_n}{d_n}\right) - 2.25\right] \tag{8}$$

where l_n/d_n is the length-to-diameter ratio of the *nth* element of the array.The center-to-center spacing s between the two rods of the feeder line, each of identical diameter d, is

$$s = d\,cosh\left(\frac{Z_o}{120}\right) \tag{9}$$

where Z_o is the characteristic impedance of the feeder line.

In the design of a log-periodic array, the directivity, input impedance, diameter of elements of feeder line, and the lower and upper frequencies of the desired operation are specified. The following design parameters are then found.

1) The optimum values of σ and τ for a given directivity are determined by using the artificial neural networks.
2) α, B_s, L, λ_{max}, N, and Z_a are determined by using eqns. (3)-(8).
3) Z_o is calculated from the design curves obtained by using artificial neural networks for various values of relative mean spacings σ' ($\sigma' = \sigma/\tau^{0.5}$).
4) s is found by using eqn. (9).

In this paper, the log-periodic dipole array were also designed by using a program called LPDA which has been written by Pozar [23]. The directivity, the radius of the largest dipole, the upper and lower frequency limits are specified, and the LPDA program computes the required number of dipoles, the dipole lengths, the dipole spacings, and the dipole radii. However, the σ and τ calculation section of the LPDA program is deleted in this study, and the values of σ and τ obtained from artificial neural networks are used in the LPDA program.

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 design of the log-periodic dipole arrays is then explained.

III. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks have many structures and architectures [5-6]. Multilayered perceptrons (MLPs) [4-6] are the simplest and therefore most commonly used neural network architectures. They have been adapted for the design of the log-periodic dipole arrays. MLPs can be trained with the use of many different learning algorithms [4-9,12]. In this work, the EDBD algorithm [12] has been used for training MLP. As shown in Fig. 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 Fig. 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.,

$$y_j = f(\sum w_{ji} x_i) \tag{10}$$

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

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.

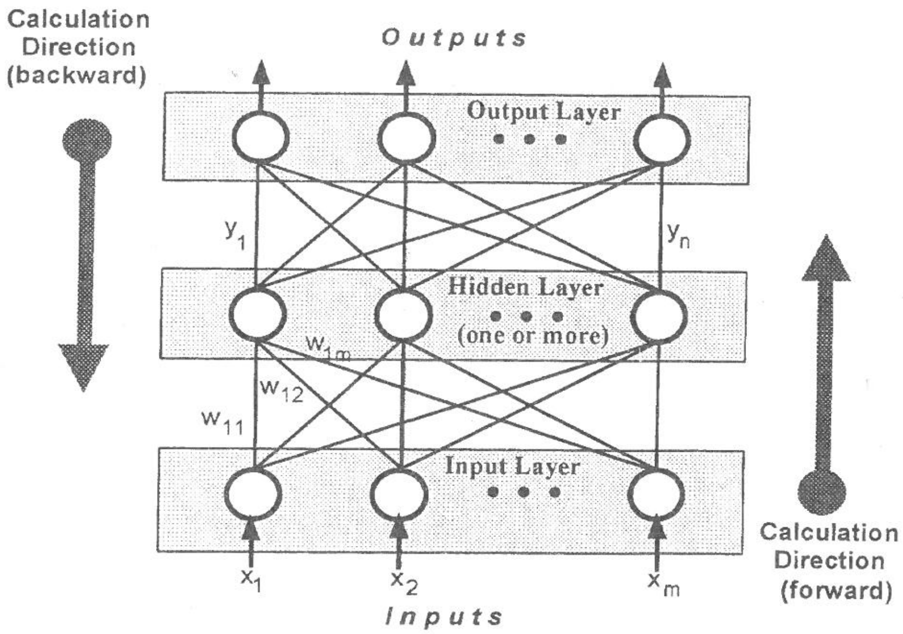


Figure 2. General form of neural networks.

III.1. Extended Delta-Bar-Delta Algorithm

This algorithm is an extension of the delta-bar-delta algorithm [9] 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 [12]. The EDBD algorithm includes a little-used 'error recovery' feature which calculates the global error of the current epoch during training [12]. 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 [7] 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

$$\Delta w(k+1) = \alpha(k)\delta(k) + \mu(k)\,\Delta w(k) \tag{11}$$

and the weights are then found as

$$w(k+1) = w(k) + \Delta w(k+1) \tag{12}$$

In eqn. (11), $\delta(k)$ 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 [9]. The weighted average $\bar{\delta}(k)$ is formed as

$$\bar{\delta}(k) = (1-\theta)\delta(k) + \theta\,\delta(k-1) \tag{13}$$

where θ is the convex weighting factor.

The learning coefficient change is given as

$$\Delta\alpha(k) = \begin{cases} \kappa_\alpha\, exp(-\gamma_\alpha\ |\bar{\delta}(k)|) & \text{if}\ \ \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_\alpha\,\alpha(k) & \text{if}\ \ \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & otherwise \end{cases} \tag{14}$$

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

$$\Delta\mu(k) = \begin{cases} \kappa_\mu\, exp(-\gamma_\mu\ |\bar{\delta}(k)|) & \text{if}\ \ \bar{\delta}(k-1)\delta(k) > 0 \\ -\varphi_\mu\,\mu(k) & \text{if}\ \ \bar{\delta}(k-1)\delta(k) < 0 \\ 0 & otherwise \end{cases} \tag{15}$$

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.(14)-(15), 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,

$$\alpha(k) \leq \alpha_{max}$$
$$\mu(k) \leq \mu_{max} \tag{16}$$

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 [7]. 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

$$E(k) > E_{min}\,\lambda \tag{17}$$

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 DESIGN OF LOG-PERIODIC DIPOLE ARRAYS

The proposed method involves training two neural networks to obtain the values of σ and τ for a given directivity and the relative characteristic impedance of a feeder line (Z_o/R_{in}) as a function of relative characteristic impedance of a dipole element (Z_a/R_{in}) for various values of relative mean spacings (σ'). These two neural models are shown in Fig. 3. In Fig. 3, R_{in} represents the real input impedance. 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 σ and τ and Z_o/R_{in} involves presenting them sequentially with different input 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 of input and output 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.

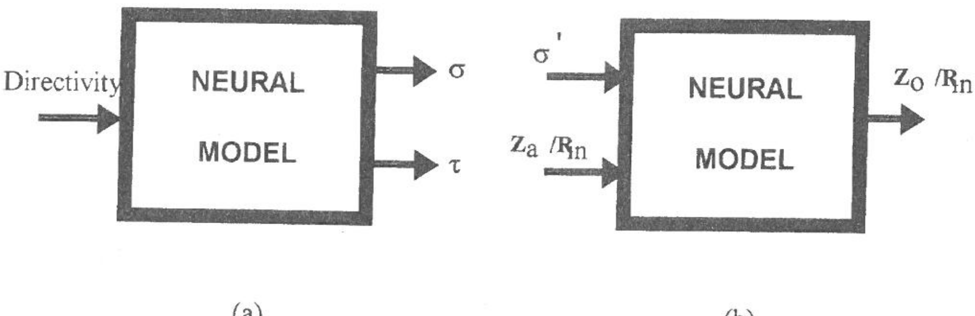


Figure 3. Neural model for (a) σ and τ calculation and (b) Z_o/R_{in} calculation.

The training and test data sets used in this paper have been obtained from the previous works [1-3]. 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. After several trials, it was found that two layers network achieved, as indicated in [10-11,13], the task with high accuracy. The number of the training and test data sets, the number of iteration for training, and the number of PEs for the first and the second hidden layers are given in Table 1. The seed number was fixed to 257. Both the sequential and random procedures were used in training. The parameters of the networks for EDBD are: $\kappa_\alpha=0.095$, $\kappa_\mu=0.01$, $\gamma_\mu=0.0$, $\gamma_\alpha=0.0$, $\varphi_\mu=0.01$, $\varphi_\alpha=0.1$, $\theta=0.7$, $\lambda=1.5$.

Table 1. The parameter values used for two neural models.

	σ and τ	Z_o/R_{in}
The number of data sets for training	32	52
The number of data sets for testing	7	18
The iteration number	200.000	250.000
The number of PEs for the first hidden layer	6	6
The number of PEs for the second hidden layer	4	3

V. RESULTS AND CONCLUSIONS

In order to demonstrate the computational effort of the neural model, the test results of ANNs for σ and τ for a given directivity and the (Z_o/R_{in}) as a function of (Z_a/R_{in}) are compared with the results of [1,3,23] in Table 2 and Fig.4, respectively. The test results illustrate that the performance of the proposed method is quite robust and precise. As can be seen from Table 2 and Fig.4, there is excellent agreement with the data from the method [1,3,23]. This excellent agreement supports the validity of ANN.

When the values of σ and τ for a given directivity and (Z_o/R_{in}) as a function of (Z_a/R_{in}) are known, the other design parameters of a log-periodic dipole array can be easily

Table 2. Comparison of the optimum test values of σ and τ for a given directivity.

	[1,3,23]		Present neural model	
Directivity (dB)	σ	τ	σ	τ
8.5	0.147	0.822	0.147191	0.822133
9.0	0.157	0.865	0.156735	0.865513
9.5	0.163	0.892	0.163481	0.893879
10.0	0.168	0.916	0.168053	0.913061
10.5	0.172	0.928	0.171471	0.928242
11.0	0.174	0.940	0.174181	0.941247
11.5	0.176	0.950	0.176348	0.952376

obtained by using eqns. (1)-(9). In this work, a log-periodic dipole array in the form of Fig.1d was designed to cover all the VHF TV channels (starting with 54 MHz for channel 2 and ending with 216 MHz for channel 13). The desired directivity is 9 dB and the input impedance is 50 ohms. The elements should be made of aluminium tubing with

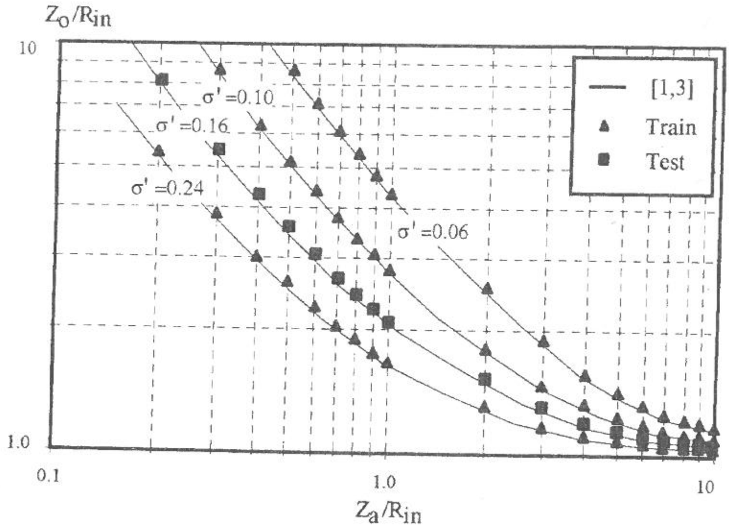


Figure 4. Comparison of relative characteristic impedance of a feeder line as a function of relative characteristic impedance of dipole element.

1.905 cm (3/4 in.) outside diameter for the largest element and the feeder line and 0.48 cm for the smallest element. These design specifications are chosen the same as those of [1] for comparison. The results obtained by using the step-by-step design procedure given in Section 2 are compared with the results of [1] in Table 3. It is clear from Table 3 that the results of ANN are in very good agreement with the results of [1].

Table 3. Comparison of the design parameter values of a log-periodic dipole array.

	[1]	Present neural model
σ	0.157	0.156735
τ	0.865	0.865513
α (°)	12.13213	12.10722
B_s	7.011224	7.000001
λ_{max}(m)	5.555556	5.555556
L (m)	5.539397	5.549253
N	14.42873 (14 or 15 element)	14.46971 (14 or 15 element)
σ'	0.1688074	0.1684726
Z_a (Ω)	327.8807	327.8807
Z_a/R_{in}	6.557615	6.557615
Z_o (Ω)	60.00	60.01
S (cm)	2.148127	2.148141

In this paper, the log-periodic array was also designed by using a program called LPDA which has been written by Pozar [23]. The specifications of the second design example are that the lower frequency is 54 MHz, the upper frequency is 216 MHz, the desired directivity is 8.5 dB and the radius of the largest dipole is 1 cm. These design specifications are the same as those of [23]. First, the values of σ and τ are obtained from the artificial neural networks. The required number of dipoles, the dipole lengths, the dipole spacings, and the dipole radii are then obtained by using the LPDA program. The results of ANN are compared with the results of [23] in Table 4. The geometry of this log periodic dipole array is also shown in Fig. 5. The spacings are measured from the current dipole to the previous dipole, so the spacing of the first dipole is taken as zero. It is seen from the Table 4 that the results of ANN are in very good agreement with the results of [23]. The two design examples given here support the validity of ANNs.

Table 4. The design parameter values of a log-periodic dipole array.

	Present neural model			[23]		
Dipole	Spacing (m)	Length (m)	Radius (m)	Spacing (m)	Length (m)	Radius (m)
1	0.0000	2.6476	0.0100	0.0000	2.6476	0.0100
2	0.7794	2.1767	0.0078	0.7784	2.1763	0.0078
3	0.6408	1.7895	0.0064	0.6398	1.7889	0.0064
4	0.5268	1.4712	0.0053	0.5259	1.4705	0.0053
5	0.4331	1.2095	0.0044	0.4323	1.2088	0.0044
6	0.3561	0.9944	0.0036	0.3554	0.9936	0.0036
7	0.2927	0.8175	0.0029	0.2921	0.8167	0.0029
8	0.2407	0.6721	0.0024	0.2401	0.6714	0.0024
9	0.1979	0.5526	0.0020	0.1974	0.5519	0.0020
σ	0.147191			0.147		
τ	0.822133			0.822		

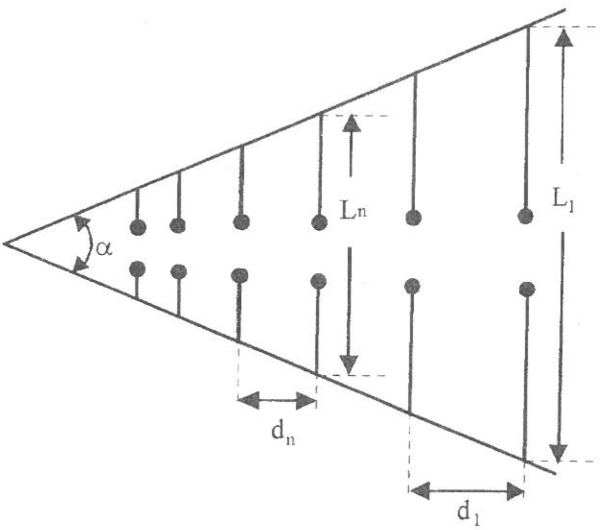


Figure 5. Geometry of a log-periodic dipole array.

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 previous our work [21], the best bandwidth results of microstrip antennas were also obtained by using the EDBD.

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. 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.

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] R.L. Carrel, "Analysis and Design of the Log-Periodic Dipole Antenna", Ph.D. Dissertation, Elec. Eng. Dept., University of Illinois, 1961.

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

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

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

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

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

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

[10] G. Mirchandani and W. Cao, "On hidden nodes for neural nets", *IEEE Transactions on Circuits and Systems*, vol. 36, no.5, pp. 661-664, 1989.

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

[12] A. A. Minai and R. D. Williams, "Acceleration of Backpropagation Through Learning Rate and Momentum Adaptation", Int. Joint Conf. on Neural Networks, Jan. 1990, Washington, DC, vol. 1, pp. 676-679.

[13] W. Y. Huang and Y. F. Huang, "Bounds on the number of hidden neurons in multilayered perceptron", *IEEE Trans. on Neural Networks*, vol.1, no.4, pp. 47-55, 1991.

[14] 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*, vol. 9, pp. 158-174, 1999.

[15] 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.

[16] 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.

[17] 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.

[18] 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), Oct. 27-30, 1998, Beijing, China, pp. 223-226.

[19] 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.

[20] 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, July 13-17, 1998, Nantes, France, pp. 57.

[21] 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.

[22] 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.*, vol. 146, pp.155-159, 1999.

[23] D. M. Pozar, "Antenna Design Using Personal Computers", Artech House, Dedham, MA, 1985.