TABLE 2. Radiation Characteristics for Different Feed Lengths of the Antenna ($h_1 = h_2 = 0.16$ cm, $\varepsilon_{r1} = \varepsilon_{r2} = 4.28$, $L = 4$ cm, $W = 2$ cm)

Feed Length S_1 (cm)	3-dB Beamwidth		Cross-Polarization (dB)	
	E-Plane	H-Plane	E-Plane	H-Plane
4	80°	70°	−27	−31
5	84°	62°	−33	−40
6	74°	70°	−33	−39

measurements, it is observed that the feed parameters have a significant effect on the radiation and reflection characteristics of the microstrip antenna. The proposed feeding technique is very simple, as compared to other methods, and offers an impedance bandwidth of 20%.

REFERENCES

1. S.H. David, A survey of broadband microstrip patch antennas, Microwave J (1996), 60–84.
2. B.L. Ooi and C.L. Lee, Broadband air-filled stacked U-slot antenna, Electron Lett 35 (1999), 51–52.
3. R.O. Lee, K.F. Lee, and J. Bobinchak, Characteristics of a two-layer electromagnetically coupled rectangular patch antenna, Electron Lett 23 (1987), 1070–1072.
4. E. Chang, S.A. Long, and W.F. Richard, Experimental investigation of electrically thick rectangular microstrip antennas, IEEE Trans Antennas Propagat AP-34 (1986), 767–772.
5. C.L. Mak, K.M. Luk, K.F. Lee, and Y.L. Chow, Experimental study of a microstrip patch antenna with an 1-probe, IEEE Trans Antennas Propagat AP-48 (2000), 777–782.
6. S. Mridula, S.K. Menon, B. Lethakumary, B. Paul, C.K. Aanandan, and P. Mohanan, Planar L-strip-fed broadband microstrip antenna, Microwave Opt Technol Lett 34 (2002), 115–117.

COMPARISON STUDY OF PATTERN-SYNTHESIS TECHNIQUES USING NEURAL NETWORKS

R. Shavit and I. Taig
Dept. of ECE
Ben-Gurion University
Beer-Sheva 84105, Israel

Received 15 December 2003

ABSTRACT: *In this paper, a comparison study among three neural-network algorithms for the synthesis of array patterns is presented. The neural networks are used to estimate the array elements' excitations for an arbitrary pattern. The architecture of the neural networks is discussed and simulation results are presented. Two new neural networks, based on radial basis functions (RBFs) and wavelet neural networks (WNNs), are introduced. The proposed networks offer a more efficient synthesis procedure, as compared to other available techniques.* © 2004 Wiley Periodicals, Inc. Microwave Opt Technol Lett 42: 175–179, 2004; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/mop.20244

Key words: *neural networks; array synthesis; wavelet neural networks*

1. INTRODUCTION

The array-pattern-synthesis problem can be defined as that of finding the array excitations to produce the required antenna-radiation pattern. Pattern synthesis is required in different applications of wireless communication, cellular communication, and radar and military systems. There are numerous synthesis methods, such as the Woodward–Lawson method [1], the Orchard–Elliott method [2], the modified Orchard–Elliott method [3], and others. Most of these methods have a high degree of complexity and sometimes they are not suitable for real-time applications. Neural networks offer a fast and easy-to-implement solution due to their unique parallel structure. The neural network is trained with input/output sets of examples with characteristics similar to those of the desired pattern. The input consists of spatial samples of the desired pattern and the output generates the array elements' excitations. After the training phase, a new pattern is introduced to the network and the array excitation is computed. In this work, the different neural networks were trained with examples generated by the Orchard–Elliott method for a flat-top pattern with constant side-lobe level (SLL).

In this paper, two neural-network algorithms for the synthesis of array patterns are introduced. The two algorithms use different activation functions, the radial basis function (RBF), and wavelets function, respectively. A comparison study was conducted among different methods: Orchard–Elliott, the multiplayer perceptrons network (MLP), the RBF network, and the wavelets neural network (WNN) network. In all three neural networks, only one hidden neuron layer was considered. The parameters for comparison are the mean square error (MSE) between the desired and computed patterns and the computation time.

2. THEORY

The basic structure of an artificial neural network is an array of processing elements (also called neurons) ordered in layers with a network of interconnection weights w_i ($j = 1, \ldots, L$) between the neurons (also called synaptic weights) [4] in different layers. The input data x_j is processed through an activation function $f(x)$. The basic model of a single neuron is shown in Figure 1. The output y of the neuron in this model is given by

$$y = f\left(\sum_{j=1}^{L} w_j \cdot x_j + b \right), \tag{1}$$

where b is the bias parameter of the activation function $f(x)$. By adding a new fixed input $x_0 = 1$ and defining the synaptic weight of this input as the bias $w_0 = b$, the bias can be evaluated in the

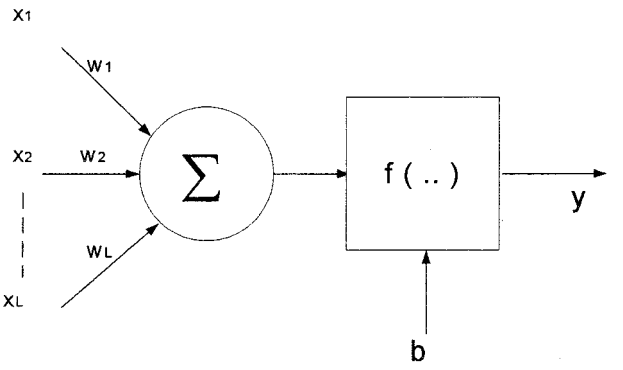

Figure 1 Basic model of a neuron

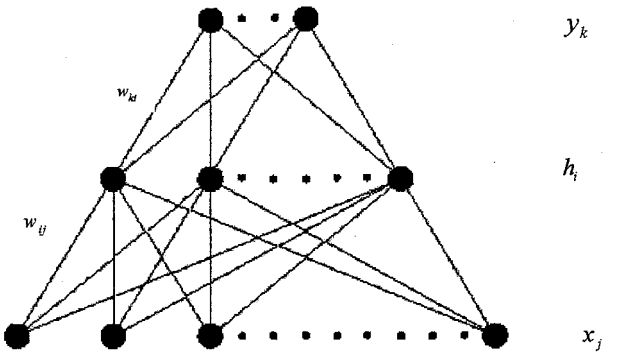

Figure 2 Schematic diagram of a neural network with one hidden layer

training stage [4]. An extension of the single neuron model to a network with input and output neuron layers interconnected with a hidden layer is shown in Figure 2. In this schematic diagram, j is the index of the input layer with $j = 1, 2, \ldots, L$, i is the index of the hidden layer with $i = 1, 2, \ldots, N$, and k is the index of the output layer with $k = 1, 2, \ldots, M$.

In the training stage, P sets of radiation patterns and excitations are used to compute the interconnection weights, w_{ij} and w_{ki}, between the layers. The interconnection weights are determined based on the requirement of minimum error between the neural model output y_k and the training data d_k. The purpose of the training process is to adjust the network interconnection weights w_{ij} and w_{ki} in order to minimize the error function $E(p)$, defined by

$$E(p) = \frac{1}{2} \sum_{k=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{L} [\, y_k(x_j, w_{ij}, w_{ki}) - d_k]^2, \tag{2}$$

where $p = 1, 2, \ldots, P$ is the index of the training set. This is an iterative process using the back-propagation algorithm described in [4–7]. The weights w_{ij} and w_{ki} are updated for each iteration by

$$\Delta w_m = -\eta \frac{\partial E}{\partial w_m}, \tag{3}$$

in which the index m stands for the indices ij or ki and η is the learning-rate parameter, which is subject to optimization (a study of this parameter is given in [5]). In this work, the learning rate was set to $\eta = 0.01$. Once the training stage is over and the weights are determined, the network is fixed such that any new input data is processed through the network in one pass from input to output.

In this work, three neural-network architectures, the MLP [4–7], the RBF [4–7], and the WNN [7–10], have been considered. In [11–13], it was shown that for large enough networks, these continuous functions can be used to determine the input-to-output nonlinear dependence of a neural network. Moreover, in [8–10], it was shown that the wavelets' activation functions are more attractive (in terms of computational efficiency) for approximating functions with sharp transitions.

The first architecture considered here is the MLP network. In this type of network, one can use one or more hidden layers. A possible activation function $f(v)$, suggested in [4–7], is given by

$$f(v) = \frac{1}{1 + \exp(-v)}. \tag{4}$$

In this case, the inputs to the network are samples of the desirable pattern. The number of input nodes (neurons) is dependent on the accuracy needed to represent the desirable pattern. The outputs of the network are the amplitude and phase-distribution values of the array elements. Accordingly, the number of output nodes should be twice the number of array elements. The layers' interconnection weights w_{ij} and w_{ki}, as shown in Figure 2, are generated in the training stage using patterns with similar characteristics to the desired pattern (for instance, generated by the Orchard–Elliott method [2]).

The second architecture considered in this work is the RBF neural network. The RBF network consists of one hidden layer of neurons, as shown in Figure 2. The RBF network output is formed by a weighted sum of the neuron outputs. The k^{th} neuron output y_k for L input neurons and M output neurons is given by

$$y_k = \sum_{i=0}^{N} w_{ki} \cdot z_i, \quad k = 1, 2, \ldots, M, \tag{5}$$

where w_{ki} is the interconnection weight of the link between the i^{th} neuron of the hidden layer and the k^{th} neuron of the output layer, and $z_i = f(\gamma_i)$ is the output of the activation RBF, where

$$\gamma_i = \sqrt{\sum_{j=1}^{L} \left(\frac{x_j - c_{ij}}{\lambda_{ij}} \right)^2}, \quad i = 1, 2, \ldots, N, \tag{6}$$

where j is the index of the input neurons $j = 1, 2, \ldots, L$, x_j denotes the network inputs, and c_{ij} and λ_{ij} are the mean and standard deviations of the radial basis activation function, respectively. c_{ij}, λ_{ij}, and w_{ki} are determined in the training process using the back-propagation algorithm, as explained previously. A commonly used activation RBF is the following Gaussian function [4–7]:

$$f(\gamma) = \exp(-\gamma^2). \tag{7}$$

The third architecture considered in this paper is the WNN network [7–10]. WNNs are feed-forward networks with one hidden layer that use a wavelet function as the hidden neuron-activation function $f(x)$. Due to the special characteristics of wavelet functions, WNNs can approximate networks patterns with sharp transitions better than MLP and RBF [8–10]. The k^{th} neuron output of the output layer with M neurons is computed using Eq. (5), where w_{ki} is the interconnection weight that controls the contribution of the i^{th} activation wavelet function to the k^{th} output, and z_i is the output of the i^{th} hidden neuron, given by

$$z_i = f(\gamma_i) = \psi\left(\frac{\boldsymbol{x} - \boldsymbol{t}_i}{a_i} \right), \quad i = 1, 2, \ldots, N, \tag{8}$$

where $\boldsymbol{x} = [x_1 \; x_2 \cdots x_L]^T$ is the input vector, $\boldsymbol{t}_i = [\boldsymbol{t}_{i1} \; \boldsymbol{t}_{i2} \cdots \boldsymbol{t}_{iL}]^T$ is the translation parameter vector, a_i is the dilation parameter, and $\psi(\,)$ is the wavelet function. A typical wavelet function chosen in this work is the inverse Mexican hat function [10], given by

$$f(\gamma_i) = \psi\left(\frac{\boldsymbol{x} - \boldsymbol{t}_i}{a_i} \right) = (\gamma_i^2 - 1) \cdot \exp\left(-\frac{\gamma_i^2}{2} \right), \tag{9}$$

where

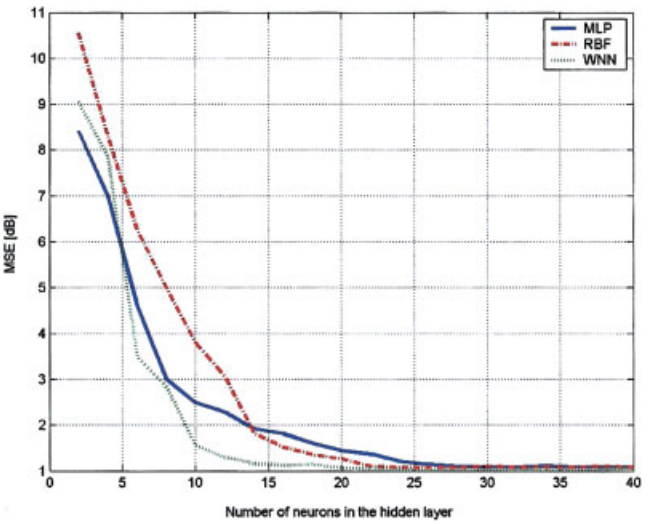

Figure 3 Dependence of the MSE on the number of neurons used in the hidden layer. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

$$\gamma_i = \left| \frac{\boldsymbol{x} - \boldsymbol{t}_i}{a_i} \right| = \sqrt{\sum_{j=1}^{L} \left(\frac{x_j - t_{ij}}{a_i} \right)^2}, \tag{10}$$

The advantages of using this wavelet function are discussed in [10]. The networks' weights, translation, and dilation parameters are obtained using the back-propagation algorithm discussed in [4–7].

3. SIMULATION RESULTS

The array used in the simulations is an 8-elements linear array with inter-element spacing $d = \lambda/2$. The training-set examples included a matrix of flat-top patterns with sector-width intervals of 20°, SLL intervals of 10 dB, and ripple intervals of 1 dB. All training patterns were generated by the Orchard–Elliott algorithm

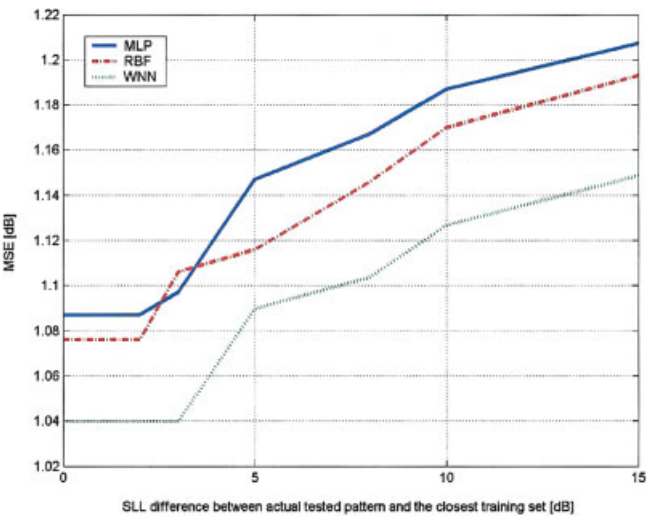

Figure 4 Dependence of the MSE on the difference in SLL between actual tested pattern and the closest training set, for a pattern sector width of 55° and a ripple of ±0.5 dB in the flat region. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

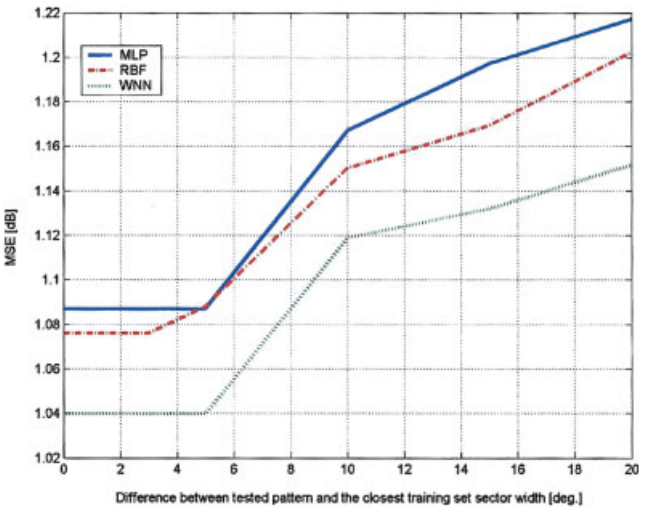

Figure 5 Dependence of the MSE upon the difference in radiation-pattern sector width between the actual tested pattern and the closest training set, for a pattern with −25-dB SLL and ±0.5-dB ripple in the flat region. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

[2]. This algorithm generates throughout the solution 2^{N1} different combinations of current distributions for each desired pattern, where $N1$ is the number of roots in the shaped region. The best solution (for practical implementation) from all different combinations is the one with minimum ratio of $|I_{\max}/I_{\min}|$. Throughout the training stage, the interconnection weights w_{ij} and w_{ki} at all levels are determined as explained in section 2. Once the weights are evaluated, the spatial samples of the desired pattern are injected in the input of the neural network and the array-element excitations are obtained at its output. The number of inputs to the network is equal to the number of samples of the desired pattern and the number of outputs is twice the number of elements in the array, since each element has an amplitude and phase excitation. Without loss of generality, the chosen test case studied in this work is the synthesis of a flat-top pattern with −25-dB SLL.

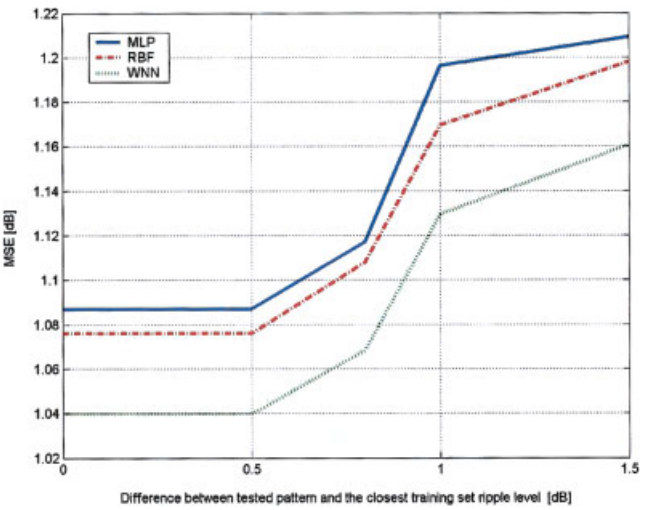

Figure 6 Dependence of the MSE upon the difference in radiation-pattern ripple between the actual tested pattern and the closest training set, for a pattern with −25-dB SLL and 55° sector width. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

TABLE 1 Comparison of the MSE and Computation Time for Different Network Architectures

Network Architecture	MSE [dB]	Computation Time [Sec]
Orchard–Elliott	1.0772	3.715
MLP	1.0871	1.1
RBF	1.0760	0.96
WNN	1.0397	0.83

Initially, a sensitivity test of the process was conducted to optimize the number of neurons N in the hidden layer. The criterion used was the MSE, given by

$$MSE = \frac{1}{Q}\sum_{q=1}^{Q}[f_q^{(c)} - f_q^{(d)}]^2, \tag{11}$$

where $f_q^{(d)}$ is the sampled desired pattern, $f_q^{(c)}$ is the sampled computed pattern, and Q is the total number of samples.

Figure 3 shows the dependence of the MSE on the number of neurons in the hidden layer. One can observe that a minimum level of 1 dB in the MSE is obtained using at least 30 neurons in the MLP network hidden layer, 24 neurons in the RBF network, and 20 neurons in the WNN network.

The sensitivity to changes in the training set was tested by using different flat-top patterns with different widths of the angular sector, SLLs, and ripple levels as training-set examples. The number of neurons in the hidden layer was taken as 30 in the MLP network, 24 in the RBF network, and 20 in the WNN network. Figures 4, 5, and 6 show the dependence of the MSE upon the difference between the desired pattern parameters (SLL, sector width, and ripple in the flat-radiation-pattern sector) to the relevant parameters in the closest training set. Figure 4 shows that for a small MSE with variation of less than ± 0.1 dB, the SLL should be in a range of up to 15 dB from the closest training set. Figure 5 shows that the pattern-sector width should be in a range of up to 15° from the closest training set for MSE variation of less than ± 0.1 dB. Figure 6 shows that the difference in the ripple of the flat-radiation-pattern sector to the ripple in the closest training set

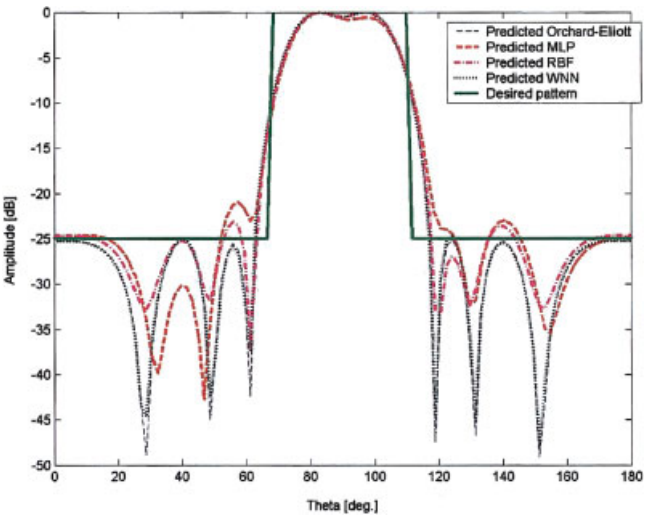

Figure 7 Comparison between the patterns synthesized by different methods for a flat-top pattern with 45° sector width, −25-dB SLL, and ±0.5-dB ripple. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

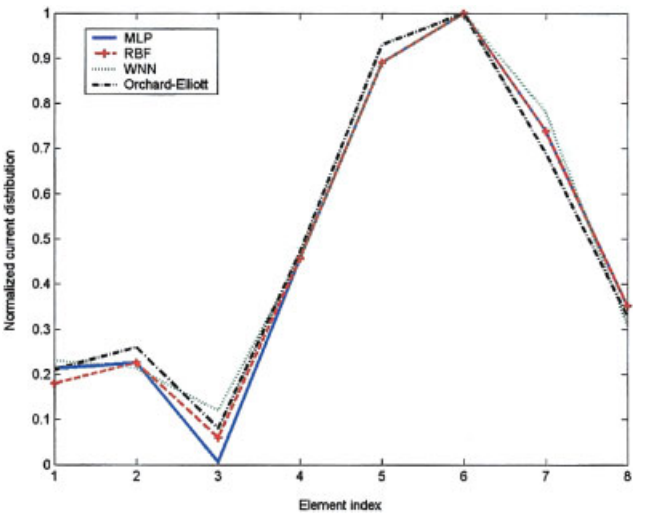

Figure 8 Comparison between the current distributions computed by different methods for a flat-top pattern with 45° sector width, −25-dB SLL, and ±0.5-dB ripple. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

should be less than 1 dB for MSE variation of less than ± 0.1 dB. In order to check the computational efficiency of the network, an arbitrary flat-top pattern (not from the training set) was introduced to the network and the output pattern was compared to the pattern generated by the Orchard–Elliott method. The MSE between the desired pattern and the actual network-output pattern was calculated and the computational time was monitored (the training phase was not included) on a Pentium-IV 1700-MHz platform. The MSE and the computational time for different networks architectures are summarized in Table 1. One can note that the WNN network has the smallest MSE error, and the shortest computational time shows an improvement of 24.5% and 13.5%, as compared to that of the MLP and RBF networks, respectively.

Figure 7 shows a comparison between the patterns synthesized by different methods for a desired flat-top pattern with a sector width of 45°, −25-dB SLL, and ripple of ±0.5 dB. It can be noted that the computed pattern using the MLP network has high side lobes (approximately −21 dB) and a slight asymmetry in the flat-top region. The SLL of the computed pattern using the RBF network is −23 dB and the computed pattern using the WNN network matches the desired pattern almost perfectly. Figure 8 shows the current distributions generated by the Orchard–Elliott algorithm, as compared to those generated by the MLP, RBF, and WNN neural-network algorithms. All current distributions are almost identical.

4. CONCLUSIONS

Three different neural-network architectures have been tested and their effectiveness for array-pattern synthesis has been compared. The results show that all three networks exhibit good performances, with preferred results given by the wavelet neural network (WNN). These techniques can be used in a vast range of applications in which the networks can be trained offline with suitable data to carry out real-time processing in order to obtain the array elements' excitations.

REFERENCES

1. P.M. Woodward and J.P. Lawson, The theoretical precision with which an arbitrary radiation pattern may be obtained from a source of finite size, Proc IEEE 95 (1948), 120–126.

2. H.J. Orchard, R.S. Eliott, and G.J. Stern, Optimising the synthesis of shaped antenna patterns, IEE Proc Microwave Antennas Propagat 132 (1985), 63–68.
3. R. Shavit and S. Levy, A new approach to the Orchard–Elliott pattern synthesis algorithm using LMS and pseudoinverse techniques, Microwave Opt Technol Lett 30 (2001), 12–15.
4. S. Haykin, Neural networks: A comprehensive foundation, Prentice Hall, New Jersey, 1999.
5. M.T. Hagan, H.B. Demuth, and M.H. Beale, Neural network design, PWS Publishing Company, Boston, 1996.
6. C. Christodoulou and M. Georgiopoulos, Applications of neural networks in electromagnetics, Artech House, Boston, 2001.
7. Q.J. Zhang and K.C. Gupta, Neural networks for RF and microwave design, Artech House, Boston, 2000.
8. Q.H. Zhang and A. Benvensite, Wavelet networks, IEEE Trans Neural Networks 3 (1992), 889–898.
9. Q.H. Zhang, Using wavelet networks in nonparametric estimation, IEEE Trans Neural Networks 8 (1997), 227–236.
10. Q. Zhang, Wavelet networks: The radial structure and an efficient initialization procedure, Tech Rpt LiTH-ISY-I-1423, Linköping University, 1992.
11. K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989), 359–366.
12. G. Cybenko, Approximation by superpositions of a sigmoidal function, Math Control Signals Syst 2 (1989), 303–314.
13. J. Park and I.W. Sandberg, Universal approximation using radial basis function networks, Neural Computat 3 (1991), 246–257.