Int. J. Electron. Commun. (AEÜ) 66 (2012) 903–912

Contents lists available at SciVerse ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.de/aeue

Neural network applications in smart antenna arrays: A review

Abhishek Rawat[a,*], R.N. Yadav[b], S.C. Shrivastava[c]

[a] Department of Electronics and Communication Engineering, Samrat Ashok Technological Institute (Degree), Vidisha, MP, India
[b] Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, MP, India
[c] T.I.T. Institute of Technology, Anand Nagar, Bhopal, MP, India

ARTICLE INFO

Article history:
Received 23 December 2010
Accepted 17 March 2012

Keywords:
Neural network (NN)
Smart antenna
Radial basis function neural network (RBFNN)
Multi-layer perceptron neural network (MLPNN)
Hopfield neural network (HNN)

ABSTRACT

Techniques employed in the synthesis of antenna arrays vary from complex analytical methods to iterative numerical methods based on optimisation algorithms. The drawback of these techniques is that they usually consider the array factor but not the interaction between array elements and real-time problems. This omission induces an error in the resultant radiation pattern; therefore, the physical relations between the array feeding details and the corresponding radiation patterns are taken into account to improve the accuracy. The behaviour of an antenna array is nonlinear in nature, resulting in an extremely high complexity using this approach, and it is usually disregarded. A neural-network-based solution can avoid complexity by establishing a relation between the desired radiation patterns and feeding details such as voltage and spacing in the real antenna array and can help convert the real array into a smart array. Several neural network applications in smart antenna array synthesis are reviewed in this paper.

1. Introduction

Smart antennas [1,2] use a fixed set of antenna elements in an array, the signals from which are combined to form a movable beam pattern that can be steered towards the user. This characteristic makes the antenna "smart" and minimises the impact of noise, interference and other issues that degrade the signal quality. The adoption of smart antenna techniques in future wireless systems is expected to have a significant effect on the efficient use of the spectrum, cost minimisation for establishing new wireless networks, optimisation of service quality and realisation of transparent operation across multi-technology wireless networks [3–7]. The concept behind smart antennas is to use base station antenna patterns that are not fixed but adapt to the current radio conditions as shown in Fig. 1. This setup can be visualised as the antenna directing a beam towards the communication partner only. Smart antenna system technologies include intelligent, dynamic phased array, digital beam forming, adaptive antenna systems [7,50,52,53]. However, smart antenna systems are customarily categorised as switched beam, dynamically phased array and adaptive array systems.

Artificial neural networks (ANN) are capable of forming arbitrary nonlinear design boundaries for complex classification tasks [8–11]. Neural networks (NN) are extremely useful in problems where the relationship between inputs and outputs is not easily modelled. An NN can approximate a model of such problems because of its ability to adapt its parameters using known input/output pairs. The NN optimises the weights between its neurons through a training process. Once the training has been completed, the network is able to interpolate results for different inputs. The growing application of ANN in a variety of electromagnetic contexts has led to their use in antenna array design and the synthesis or optimisation of radiation problems. The high-speed capabilities and learning abilities of NN can be applied to solve numerous complex optimisation problems in electromagnetic and antenna arrays. The inherent nonlinearities associated with antenna radiation patterns make antennas suitable candidates for NN. The authors also identify some situations where the use of NN is the best problem-solving option in electromagnetic and antenna design [12,13].

In Section 2, different NN suitable for smart antenna are discussed. Section 3 presents multi-layer perceptron (MLP)-based smart antenna array applications in fault-finding and null enforcement. Radial basis function (RBF)-based and Hopfield neural network (HNN)-based applications are presented in Sections 4 and 5, and hybrid and wavelet NN-based smart antennas are discussed in Section 6. Section 7 concludes the paper.

* Corresponding author at: Warden Quarter of I.G. Hostel, SATI Campus, Samrat Ashok Technological Institute (Degree), Vidisha, MP 464001, India.
Tel.: +91 07592 250346.
E-mail addresses: arawat@satiengg.org (A. Rawat), yadavrn@manit.ac.in (R.N. Yadav), scshrivastava@manit.ac.in (S.C. Shrivastava).

1434-8411/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2012.03.012

Table 1
Summary of various MLPNN applications in antenna arrays.

Application	I/P neuron-hidden neuron –O/P neuron	Network training	Training time	Learning rate	Success rate
Determining the number and location of the fault elements in the array in space platforms and locating a maximum of three defective elements in a 16-element array [15]	37-15-3	MLP is trained in backpropagation mode	(on a 66 MHz P4) 25 min	0.05	–
Finding failed element positions in linear antenna arrays [19]	40-15-5	MLP is trained in the backpropagation mode	(on a 66 MHz P4) 14 min	0.05	–
Estimating DOA in linear antenna array [28]	21-4-2	Multi-layered feed-forward neural network	–	–	–
Finding defective elements [17]	50-180-100	MLP is trained in backpropagation mode	–	–	76.7%
Null enforcing in field pattern using the interpolation capacity of NN [45]	121-100-80-52	Resilient backpropagation algorithm	–	–	96.57%

2. Neural networks suitable for smart antenna applications

The ANN can be successfully used in different smart antenna array (SAA) applications in communication systems. The use of SAA in communication systems not only improves system performance but also extends its functions and hence its broad applications in communication systems.

In general, MLP-, RBF- and Hopfield-type NN are the most suitable for use in various SAA applications. In the NN design, we do not have any prior knowledge of the number of neurons needed in the hidden layer. The inclusion of too many neurons in the hidden layer increases the computational complexity and processing time; however, too few neurons would increase the classification error. Therefore, determining the appropriate number of neurons in the hidden layer is one of the most critical tasks in NN design. Certain characteristics of the NN must be defined before its use. First, an adequate structure must be chosen for the network and then trained and tested with a broad dataset for required applications. The MLP-, RBF- and Hopfield-type NN provide excellent results in various applications of SAA. For some specific problems, hybrid techniques such as ANN and genetic algorithm (GA) or source reconstruction methods also provide good results [46,47]. Wavelet NN can also be used to consider the mutual coupling effect during the array synthesis.

3. MLP-based smart antenna arrays

MLPNN, one of the most common neural networks, has an architecture that may contain two or more layers, as shown in Fig. 2. The input layer is the first layer in which the number of its neurons is equal to the number of selected specific features. The output layer is the last layer, which determines the desired output classes. The intermediate hidden layer may increase the MLPNN's capability and is most useful for nonlinear systems. An MLPNN trained in backpropagation mode can be used to locate the fault elements in an antenna array, and multilayered feed-forward allows the estimation of the arrival direction. Table 1 summarises all MLPNN applications in antenna arrays.

3.1. Fault-finding in antenna arrays

The antenna arrays used in sonar, radar and other communication applications contain several hundred radiating elements, so the failure of one or more elements is a reasonable scenario and must be addressed. Element failure in antenna arrays destroys symmetry and causes unacceptable pattern distortion. This problem can be solved by replacing the defective elements in aircraft antennas but remains a critical problem in space platforms. Several compensation techniques have been reported in the literature [14,17]. To apply these compensation techniques, the number and position of the faulty elements in the array must be known beforehand. Active antennas include calibration systems to identify the damaged element or elements. These systems provide an easy way to control the system components, but this control fails if the calibration system is also damaged. This problem can be solved by an MLP trained using a backpropagation algorithm [15]. Element failures cause sharp variations in the field intensity across the array

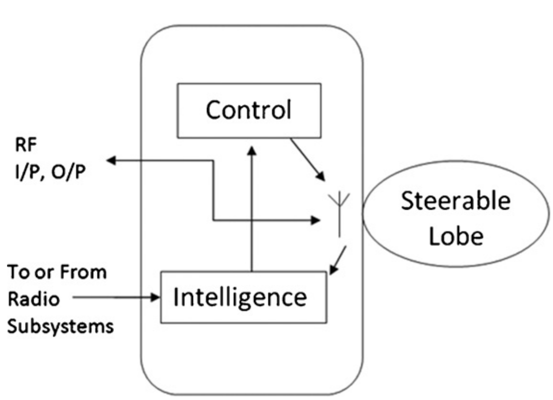

Fig. 1. Concept of smart antennas.

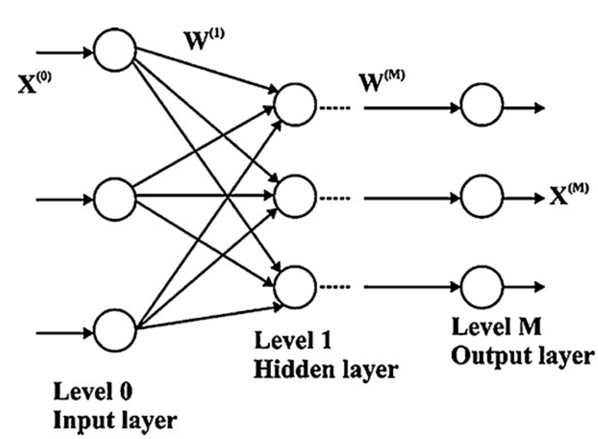

Fig. 2. MLPNN architecture.

aperture, increasing both the side lobe and the ripple level of the power pattern. From the viewpoint of a faulty element, the number and size of the side lobes and radiation pattern are a function of the number and position of the faulty elements in the array. In this approach, the faulty elements are treated as completely non-radiating, but the presence of the faulty element in the array has its own effect on the overall radiation pattern. With the increase in number of faulty elements, the variation in the radiation pattern also increases. The task here is to create a mapping between the distorted radiation patterns and the corresponding location of the faulty elements in the array.

The MLPNN technique can be successfully used to determine the position of the faulty elements in the array. The NN creates a mapping between the damaged radiation patterns and the position of the faulty elements in the array. The MLPNN uses the radiation pattern as its input and provides the position of the faulty elements as output. This proves the effectiveness of MLPNN technique, in the absence of an experimental facility. To generate training data for the MLP, the simulated radiation patterns for the random location of non-radiating elements is generated with one, two and three element faults, and the patterns are sampled. Taking samples of the radiation pattern is also justified because it is not possible to know the entire damaged pattern at the base station for the antenna arrays in the space platforms. In this approach, the MLP is trained in backpropagation mode because of its suitability for creating mapping implementations. In this training algorithm, the weight updating occurs according to

$$w_i^{k+1} = w_i^k - \eta \frac{(\partial E^k)}{(\partial w_i)} \tag{1}$$

where η is the learning rate and E^K is the mean square error (MSE) at the Kth instant. The NN learns from samples of input/output data, i.e., a_k and b_k; $k = 1, 2, \ldots N$, where N is the total number of samples for training (number of fault combination). In this procedure, the performance function of the feed-forward network is modified by adding a term with the MSE that consists of the mean of the sum of the squares of the network weights and biases. The performance function [16] now becomes

$$E = \gamma \left[\frac{1}{N} \sum_{i=1}^{N} (t_i - a_i)^2 \right] + (1-\gamma) \left[\frac{1}{N} \sum_{j=1}^{n} w_j^2 \right] \tag{2}$$

where γ is the performance ratio, t_i is the target output, a_i is the network output of ith neuron, and n is the total number of weights (w) and biases. Rodriguez et al. [17] also compared several techniques that allow the determination of the number and location of the defective elements in arrays, such as the Woodward–Lawson method [20], NN and GA [21], and found that the Woodward–Lawson method achieves good results at the expense of requiring many measurements of both the amplitude and phase of the degraded pattern. The applicability of this method in a real case is therefore limited. The GA also achieves good results even with measurement errors, but its computational cost is high. A case-based reasoning system can significantly decrease this cost by reducing the number of combinations of defective elements. NN would be useful only to select a set of candidates for defective elements as a stage before the GA. The benefit of using ANN is that it completely bypasses the iterative process when equipped with a properly trained network. In addition, the GA technique requires both the initial radiation pattern and the base station's measurement of the damaged radiation pattern. The author of [23] uses ANN to create a mapping between the damaged radiation patterns and the position of the faulty elements in the array that is accomplished using an MLP trained in the backpropagation mode.

Table 2
Summary of Various RBFNN applications.

Applications	Training	Neuron
Digital beamforming [27]	k-means algorithm	200
Detection and direction of arrival (DOA) estimation [29]	N-MUST algorithm	72, 156
Direction finding [35]	Trains using a least mean squared (LMS) error solution.	–
Obstacle modelling	k-means algorithm	The number of inputs is twice the number of samples of the radiated field distribution
Finding defective elements	Backpropagation	50-[] –100

3.2. Null enforcement in field pattern

The MLP architecture can be successfully used to reject some interference in the coverage area of the antenna array radiation pattern if its position is established. The author in [45] used the interpolation capacity of MLP for the null enforcement in field pattern to generate a quasi-null in the global beam-radiation diagram of an array installed in a geostationary satellite. The architecture of the MLP used allows the rejection of some interference in the covering zone of a planar array's radiation pattern when the direction of interference has been established. The MLPNN is an overlooked way to obtain the inverse equation of the radiation pattern of a planar antenna. This technique could be used to further investigate diagnosing array faults using measured values of the array's far-field amplitude [22], input impedance [24] and mutual coupling [25]. Thus, MLPNN would be a very useful method for antennas permanently or semi-permanently deployed in space and for any condition that requires the use of costly near- and far-field measurement facilities.

4. RBFNN-based smart antenna arrays

RBFNN [23] is general-purpose method for approximating non-linear mappings. RBF neurons can be trained faster than MLP because of its two-stage training procedure. RBFNN can be trained with supervised and unsupervised learning and provides excellent results for obstacle modelling, direction of arrival (DOA) and beamforming. The first layer contains the parameters of the basic function, while the second layer forms linear combinations of the activation of basic functions to generate outputs. Unlike back-propagation networks, which can be considered an application of an optimisation problem, RBFNN can be considered a curve-fitting problem in a high-dimensional space. Table 2 summarises all RBFNN applications in antenna arrays.

4.1. Obstacle modelling in array synthesis

Real radiating systems do not work in free space conditions. Obstacles, the environment or the mechanical structure of an antenna can modify its properties so that the results of a synthesis process are not perfectly valid. The direct way to consider this effect is to introduce a model of the radiating system with real-time conditions. To compensate for the effect of the obstacle on the radiation pattern generated by an antenna, the coupling effects between the radiating structure and the obstacles between the array elements must be accounted for. These coupling effects cannot be handled using traditional synthesis methods, including those based on ideal excitations or array factor coefficients [26]. Traditional approaches for synthesis are based on different approximations, such as the array factor formulation, that imply that there is no interaction between the radiating elements. In a real radiating system, coupling effects between elements of the array modify their individual radiation properties and consequently the global radiation pattern. If these effects are not considered, a certain error is assumed in the solutions. If P sets of voltages with random amplitudes and phases are generated, v_i, $i = 1, 2 \ldots P$, with $v_i = [v_{i1}, v_{i2}, \ldots v_{iL}]$, L being the number of feeding ports, the corresponding radiated field distributions, $(\bar{E}_i(x, y, z))$, can further be calculated using an analysis-tool-based method of moments (MoM) and can be expressed using an unknown analysis operator A as

$$\overline{E_i}((x, y, z) = A(V_i)) \quad (3)$$

The purpose of the synthesis problem is to calculate a function A^{-1} so that $v_i = A^{-1}$ $(\bar{E}_i(x, y, z))$. NN can approximate such a model because of its ability to adapt its parameters using known input/output pairs. The NN optimises the weights between its neurons through a training process. Once the training has been properly completed, the network is able to interpolate results for different inputs. This ability of NN can be used to relate a set of voltages applied in ports of an antenna array and the resulting radiated field distribution. P patterns $\{v_i(\bar{E}_i(x, y, x))\}$ are then used to train the NN. For simplicity, feed-forward NN have been considered, and RBFNN show better properties that MLP for synthesis purposes. Although both structures are universal approximations [23], the local response of the neurons in an RBFNN makes it more suitable for the synthesis problem. Theoretically, an MLP should be able to perform synthesis with the same accuracy as an RBFNN provided given enough neurons, but in practice, too many neurons could overflow many applications. Once the network structure has been chosen, the number of neurons must be defined. Although some rules have been proposed to choose the number of neurons that must be used in an NN, none of them is valid in a general case, and only experimental validation of synthesis results can assure the desired accuracy.

Rafael et al. [26] developed an NN-based synthesis method that can obtain the voltages that must be applied to the elements of a given array, considering coupling effects between elements and the presence of a nearby conducting obstacle.

4.2. One- and two-dimensional antenna array designs

The RBFNN approach can be used to find the weights of one- (1-D) and two-dimensional (2-D) antenna arrays. In modern cellular satellite mobile communications systems and global positioning systems, both desired and interfering signals change their directions continuously. Therefore, a fast tracking system is needed to constantly track the users and then adapt the radiation pattern of the antenna to direct multiple narrow beams to the users and null interfering sources. In this approach, the computation of the optimum weights can be accomplished using three-layer RBFNN [27].

The results obtained from this network are excellently matched with the Wiener solution. The antenna array can detect and estimate mobile users' locations, track them as they move within or between cells, and allocate narrow beams in the directions of the desired users while simultaneously null to unwanted sources of interference. This SAA results in an increased system capacity for the existing cellular and mobile communications systems. For a linear array of M elements, we can write the array output on the M dimensional vector as

$$X(t) = A \cdot S(t) + N(t) \quad (4)$$

where A is the steering matrix and $S(t)$ is the vector of the received signal at any time t. For a general $M \times N$ rectangular array, the received signal data can be arranged in a $1 \times MN$ vector given by

$$X(t) = \sum_{i=1}^{K} S_i(t) \cdot A_i + N(t) \quad (5)$$

where $N(t)$ represents the noise. To derive the optimal weight vector, the array output is minimised so that the desired signals are received with a specific gain, while the contributions due to noise and interference are minimised. This process yields

$$W_{opt} = R^{-1} S_d \left[S_d^H R^{-1} S_d \right]^{-1} \cdot r \quad (6)$$

where R is defined as $E\{X(t) \cdot X^H(t)\}$, S_d is the steering matrix pointing to the desired signals given by

$$S_d = \left[S_d(\theta_1), S_d(\theta_2), \ldots S_d(\theta_v) \right] \quad (7)$$

and $S_d(\theta_i)$ is defined as

$$S_d(\theta_i) = \left[1, e^{-jki}, \quad e^{-j2ki}, \ldots e^{-j(M-1)ki} \right] \quad (8)$$

where r is the $V \times 1$ constraint vector and V is the number of desired signals. The performance phase of RBFNN can be accomplished by generating the array output vector $X(t)$, normalising this array output vector by its norm and presenting the normalised array output vector at the input layer of the trained RBFNN. The output layer of the trained RBFNN will produce an output that estimates optimum weights for the array outputs. Unlike the least mean square (LMS), recursive least squares (RLS), or the sample matrix inversion (SMI) algorithms, where the optimisation is carried out whenever the directions of the desired or interfering signals change, the weights of the trained network can be used to produce the optimum weights needed to steer the narrow beams of the SAA in real time in this approach.

4.3. ANN approach for direction finding

Standard antenna beam forming algorithms, such as monopulse, require calibrated antennas because they depend on nearly identical antenna element performance. These algorithms do not perform well with physically untested antennas or unknown degradations. As phased-array antennas become larger and more highly integrated into physical structures, this uniformity requirement generates production and maintenance costs that are increasingly prohibitive for many military and civilian applications. The ANN approach can solve this problem.

The neural beam former architecture consists of antenna measurement input preprocessing, an artificial neural network and output postprocessing. Network preprocessing exploits antenna expertise to simplify and enhance NN inputs. It removes irrelevant information, eliminates artificial discontinuities in the input function space and reduces the inputs to a small set of relevant information. Network preprocessing can be trained if their locations are known and boundary points are available. These techniques usually create larger and slower networks than those utilising intelligent preprocessing [29]. Therefore, NN architecture that has proven successful at approximating continuous functions from a small set of samples and can be trained across discontinuities should be chosen. Thus, as the number of inputs increases (larger antenna), the size of the network layers should grow at the same rate with no additional layers required. The RBF network architecture reportedly satisfies all of these constraints. The architecture of a three-layer RBF network, shown in Fig. 3, consists of an input layer, a hidden layer of Gaussian RBFs, and an output layer of summation nodes. The input nodes receive the preprocessed antenna data and broadcast the input vectors to each hidden layer node.

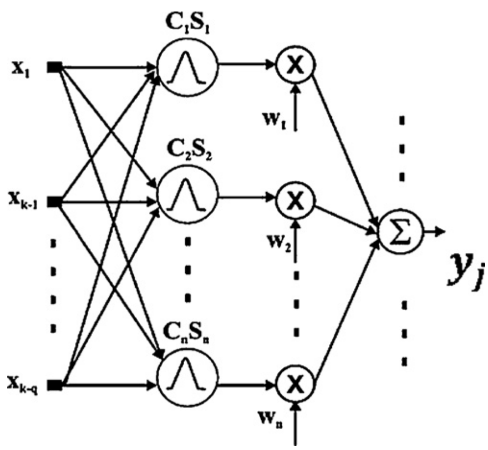

Fig. 3. RBFNN architecture.

Each input vector x is an element of the n-dimensional input space $\tilde{x}$. For these n-component input vectors, $x=(x_1,x_2,x_3\ldots x_n)$, preprocessing ensures such that $x_k \in [-1,1]$ where $k=1,2,3,\ldots,n$. The hidden layer maps $\tilde{x}$ into a space $\tilde{\varphi}$ that consists of q-component Gaussian RBF vectors φ, where q is the number of hidden layer nodes.

$$\varphi_i = \exp\left[-\sum_{k=1}^{n}\frac{(x_k-m_{ik})^2}{2\sigma_i^2}\right] \tag{9}$$

where φ_i is the output of the ith hidden layer node for $i=1,2,3,\ldots q$ and φ_i is calculated from the input vector x, the Gaussian centre m_i and the spread parameter σ_i. Initially σ_i is a trainable parameter that varied for each Gaussian node. Each output node computes a weighted sum of the outputs generated by the hidden layer nodes [29].

$$y_j = \sum_{i=1}^{q}\varphi_i\omega_{ji} \tag{10}$$

where ω_{ji} are the connecting weights. Thus, the output layer maps $\tilde{\varphi}$ into $\tilde{y}$ the space of all possible angular directions to the source, where the output vector $y=(y_1,y_2\ldots y_r)\in\tilde{y}$ for r output nodes. The energy in the r output nodes is then postprocessed to estimate the source angle of arrival. For single-source direction finders (DFs), we can train a single output node to emit a value proportional to the source angle. After the learning process, the network can predict antenna behaviour at points between the training points by generalisation. Comparing the two variations of the RBF neural network, an adaptive network, ARBF, which uses gradient descent optimisation training, and a linear-algebra-based network, (Linnet), which trains using a least mean squared (LMS) error method. Comparisons between ARBF and Linnet and analysis of an error-weight surface show that the ARBF implementation converges to a near-optimal solution in a few iterations.

4.4. Direction of arrival estimation

The direction of arrival (DOA) estimation of mobile users using linear antenna arrays is very complex and needs a very large number of calculations. To reduce the computational complexity of super-resolution algorithms, e.g., multiple signal classification (MUSIC), the DOA problem is approached as a mapping, which can be modelled using a suitable ANN trained with input/output pairs. The three-layer RBFNN can learn and provide an appropriate solution for multiple source-direction findings of multi-element arrays. The network weights can be modified using the normalised cumulative delta rule. The performance of this network is compared with that of the MUSIC algorithm for both uncorrelated and correlated signals. The RBFNN substantially reduces the CPU time for DOA estimation computations [31].

Various DOA estimation methods exist [30], some of which allow the direction to be estimated from the spatial-frequency spectrum, which can be obtained using a discrete Fourier transform (DFT) [33]. These spatial-spectral estimation algorithms can be used to obtain optimum or sub-optimum weights using spatial samples at one time instant. Therefore, if the processing speed is fast enough to track the time variation of channels, these algorithms can be more attractive for a fast-fading channel than the temporal updating algorithms. Most recently, NN have also been used to detect the DOA. Once the NN is trained offline, multiple signals can be tracked in real time [32,34].

4.5. Multiple source tracking smart antenna

The multiple-source tracking problem can be solved using NN-based smart antennas for wireless terrestrial and satellite mobile communications. The neural multiple-source tracking (N-MUST) algorithm is based on the architecture of a family of RBFNNs to perform both detection and DOA estimation [28]. The field of view of the antenna array is divided into spatial angular sectors, which are in turn assigned to a different pair of RBFNNs. When a network detects one or more sources in the first stage, the corresponding second-stage networks are activated to perform the DOA estimation. The weight coefficients are updated using the Wiener method, derived from the estimated spatial spectrum. Moreover, the MUSIC algorithm estimates the DOAs in the noise [26]. This approach is based on dividing the field of view of the antenna array into angular spatial sectors and then training each network in the first stage of the algorithm to detect signals emanating from sources in that sector. Once this first step is performed, one or more networks of the second stage (DOA estimation stage) can be activated to estimate the exact location of the sources [31,33].

For a linear array composed of M elements, let K ($K<M$) be the number of narrowband plane waves, centred at frequency ω_0 impinging on the array from directions $\{\theta_1,\theta_2,\ldots\theta_K\}$. Using complex signal representation, the received signal at the ith array element can be written as.

$$X_i(t) = \sum_{m=1}^{K} S_m(t)e^{-j(i-1)k_m} + n_i(t) \quad \text{and} \quad i=1,2,3,\ldots\ldots m \tag{11}$$

where $S_m(t)$ is the signal of the mth wave, $n_i(t)$ is the noise signal received at the ith sensor and

$$k_m = \frac{\omega_0 d}{c}\sin(\theta_m) \tag{12}$$

where d is the spacing between the elements of the array, and c is the speed of light in free space. Using vector notation, we can write the array output in matrix form:

$$X(t) = AS(t) + N(t) \tag{13}$$

where $X(t)$, $S(t)$, and $N(t)$ are given by

$$X(t) = [X_1(t), X_2(t), X_3(t), \ldots X_M(t)]^T$$

$$N(t) = [N_1(t), N_2(t), N_3(t), \ldots N_M(t)]^T$$

$$S(t) = [S_1(t), S_2(t), S_3(t), \ldots S_M(t)]^T$$

The superscript "T" indicates the transpose of the matrix. Moreover, A is the $M \times K$ steering matrix of the array towards the direction of the incoming signals defined as

$$A = \left[a(\theta_1), \ldots a(\theta_m), \ldots a(\theta_k)\right] \tag{14}$$

where $a(\theta_m)$ corresponds to

$$a(\theta_m) = [1, e^{-jk_m}, e^{-j2k_m}, \ldots e^{-j(M-1)k_m}]$$

Assuming that the noise signals $\{n_i(t), i = 1{:}M\}$ received at the different sensors are statistically independent white noise signals of zero mean and variance σ^2 and are independent of $S(t)$, the received spatial correlation matrix R of the received noisy signals [33] can be expressed as

$$\begin{aligned} R &= E\{X(t)X^H(t)\} \\ R &= AE[S(t)S^H(t)]A^H + E[N(t)N^H(t)] \end{aligned} \tag{15}$$

In the above equation, "H" denotes the conjugate transpose. The antenna array can be considered to be performing a mapping $G{:}R^K \to C^M$ from the space of the DOAs, $\{\Theta = [\theta_1\, \theta_2 \ldots \theta_K]^K\}$ to the space of sensor output $\{X(t) = [x_1(t), x_2(t), x_3(t), \ldots x_m(t)]^T\}$. The NN can successfully perform this inverse mapping $F{:}C^M \to R^K$ for multiple source tracking.

4.6. Array synthesis with mutual coupling effect

Traditional synthesis methods usually address the array factor information using approaches based on considering ideal radiating elements without interaction between them. Accounting for the coupling between elements increases the complexity of the synthesis tasks, so it is usually disregarded assuming a certain error in the solution. This error can be unacceptable in some antenna array designs. An NN-based solution can exploit the "a priori" knowledge of the radiating system to relate a given radiated field distribution to the voltages that must be applied to each radiating element, taking into account mutual coupling effects without increasing the complexity from the designer's perspective.

The synthesis problem in array design consists of finding the excitation distribution of the antenna elements for given radiation pattern with specified characteristics. If the analysis problem is considered the calculation of the radiation pattern from a given excitation law using analytical or numerical tools, the synthesis problem can be considered the inverse of the analysis problem, that is, the calculation of the excitation distribution that generates a given field distribution. Elementary antennas used as radiating elements in an array present mutual coupling effects depending on their geometry and separation such that each element induces currents over the others, modifying their individual radiation pattern and their contribution to the global radiated field distribution. Traditional approaches to the synthesis problem are usually based on direct analysis of a set of coefficients from the array factor or other analytical representations of the radiation pattern [25,44,57]. Other recent techniques use cost functions involving the desired field pattern and the field expressed in terms of the array factor or integral equations including the equivalent currents over a surface "enclosing" the antenna [58]. An advantage of using this type of cost function is that it allows the inclusion of added constrains [59]. Some regularisation and optimisation methods have been proposed in [60,61] to improve the ill-conditioned behaviour of the numerical solution of the cost function and to obtain an accurate estimation of the excitation law through its minimisation. Under certain circumstances, the values of the array factor coefficients can be considered proportional or directly related to the voltages that must be applied to the ports of the array without accounting for the coupling effects between elements once inserted in the array, but this approximation assumes a certain degree of error that is impractical for some applications. An NN-based synthesis tool can avoid all of these problems, taking into account the coupling effects and the real radiation properties of the structure under study without any increase in complexity.

The authors of [48] proposed a set of different array synthesis techniques based on NN with radial basis activation functions that used mutual coupling between elements and addressed different methods of pattern specification (amplitude and phase, amplitude-only data, template-based specifications). Deterministic pattern specifications suitable for the MSE criterion are given in Eq. (16). This method is not appropriate when a template with maximum and minimum values of the radiation pattern at every aspect angle is specified. A procedure for phase handling when the feeding amplitude values must be kept constant is also proposed.

$$\mathrm{MSE} = \frac{-1}{2P}\sum_{i=0}^{P}|y_i^d - y_i^a|^2 \tag{16}$$

where y_i^d is the desired pattern and y_i^a is the actual pattern for different range of P elements.

4.7. Non-uniform-antenna array synthesis

The ANN can relate the radiation pattern to the voltages that must be applied to each radiating element without making any assumption about the antenna geometry. This approach can solve non-uniform-antenna array problems, even considering coupling effects. The authors in [49] proposed an RBFNN model for the synthesis of a linear array of eight different sized dipoles non-uniformly distributed and two half-wavelength passive dipoles placed at the ends of the structure. The advantage of NN-based synthesis is its extension to non-uniform-antenna array synthesis. The set of voltages applied to the ports of an antenna array is related to the field distribution radiated by the structure. The NN can accurately approximate the unknown function A^{-1} through a learning process using generated input/output pairs using the function A. This concept leads to a new approach to the problem in which no particular geometry need be considered. If the antenna can be analysed using an efficient tool able to account for the real properties of the structure, a set of training pairs can be generated. This set can be used to adapt the internal parameters of an NN. This NN will then be able to perform synthesis tasks for arbitrary geometry arrays.

5. HNN-based smart antenna arrays

The architecture of HNN as shown in Fig. 4 is a single-layered, fully connected NN with symmetric interconnections. This NN has an associated energy function that the network seeks to minimise. This feature can be exploited to solve optimisation problems, such as minimising the MSE between a reference signal and the received signal. The HNN has been used to implement adaptive beam forming and tracking. A comparison between the performance of the

Table 3
Summary of various HNN applications.

Applications	Type	Key feature
Neural tracker for phased array antenna [43]	Single layer, feedback	Powerful computation, easy implementation
Adaptive beam forming with direct sequence spread spectrum [37]	Single layer, feedback	Powerful computation, easy implementation

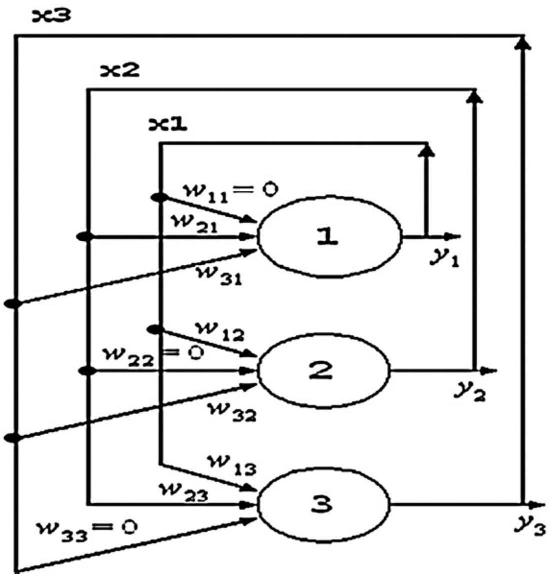

Fig. 4. HNN architecture.

Hopfield algorithm and the LMS algorithm in [36,37] indicates that the Hopfield beamformer outperforms the classical LMS algorithm in terms of convergence rate. Table 3 summarises all HNN application of antenna array.

5.1. Neural tracker for phased array antenna

A neural state estimator based on the Hopfield network can be used for phased array tracking. The neural tracker uses dynamic optimisation to perform state estimation in a sequential manner and does not require prior statistical information on the noise [38,39]. The update time can also vary randomly depending upon the operator, which cannot be achieved with track-while-scan radar because of its severe limitations by mechanical inertia. Therefore, it is more appropriate to include the variable update time in the algorithm for target tracking with a phased array. The HNN state estimator for phased array provides excellent results when using the combination variable update time concept.

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + n(t) \\ z(t) &= C(t)x(t) + w(t) \end{aligned} \tag{17}$$

where $x(t)$ is a $m \times 1$ state vector, $A(t)$ is an $m \times m$ known state transition matrix and $n(t)$ is an $m \times 1$ known system error vector, $z(t)$ is a $k \times 1$ observation vector, $C(t)$ is a $k \times m$ known observation matrix, $w(t)$ is a $k \times 1$ observation error vector. To obtain an optimal state estimator of the above system, a combined MSE objective function given as

$$F = \frac{1}{2}\int_{t_0}^{t_f} \left[(z(t) - c(t)x(t))^T Q((z(t) - c(t)x(t)) + n(t)^T Rn(t)\right] dt \tag{18}$$

is used [39]. The time period under consideration is $[t_0, t_f]$. Q and R, the estimator design parameters, are $k \times k$ and $m \times m$ symmetric positive definite matrices, respectively. This optimisation problem is well defined because the integrand is a positive function for any $x \in$ real [43].

5.2. Adaptive beam forming with direct sequence spread spectrum

Adaptive beam forming is a robust technique for improving the reception of the desired signal and protecting the communication systems from interference signals. This technique involves the formation of multiple beams, with the main beam steered towards the desired signal, other beams towards the multi-path signal, and nulls in the direction of the interfering signals [40]. This outcome is achieved by weighting each received signal according to an adaptive algorithm. When such a technique is cascaded with a spread spectrum system, it exhibits a greater capacity for suppressing unwanted signals than does a classical beam former.

An HNN can be used to perform adaptive beam forming with direct sequence spread spectrum signals. A reference signal generator is used to derive the desired signal for the adaptive processor. A PN code is used to spread the data signal, which is bi-phase modulated. At the receiver end, this code is used to extract the desired signal component from the array output and reject any narrowband interference signals. The communications literature abounds with adaptive algorithms for beam forming. NN can provide very attractive solutions to complicated problems because of their highly interconnected structure, computational efficiency and rapid convergence. A Hopfield network was used in [41] to implement adaptive beam forming for a spread spectrum communications system. One can also integrate a Hopfield neuro beam former [42] into a spread spectrum system along with a reference signal generator. This system results in automatic tracking of the desired signal and suppression of interfering signals.

6. Hybrid techniques and wavelet neural networks

6.1. ANN and source reconstruction methods

The drawback of traditional array synthesis techniques is that they usually work with the array factor information, without considering the interaction between antenna elements. This omission causes a certain error in the resulting radiation pattern, and desired specifications might not be fulfilled. The accuracy can be improved by considering the physical relations between the dipole array voltages and the corresponding radiation patterns. This approach is extremely complex and is usually disregarded. An NN-based solution can avoid complexity by establishing a relationship between the desired radiation pattern and voltages in the real antenna. In most practical cases, specifications are given as a set of far-field radiation (FF) parameters or templates. NN are not able to handle so many degrees of freedom without any further constraint, but a combination of techniques can turn specifications into a certain equivalent near-field radiation (NF) and then into voltages. A three-stage method was proposed in [46]. The first stage is an NN training procedure that uses a full-wave analysis method based on the method of moments (MOM) and wire modelling (WM) such that the NN can relate NF patterns to the voltages that produced them. In the second stage, an equivalent source reconstruction method is used to obtain a set of electric or magnetic currents able to produce a radiation pattern complying with the specifications. These currents are used to calculate the NF pattern values. In the third step, trained NNs provide output from previously calculated NF data [46], as shown in Fig. 5.

This method works directly on design parameters without any simplification of the properties of the radiating elements. Therefore, no approximation is assumed, and the solutions are highly accurate. Additionally, the use of voltages instead of ideal excitation parameters or physical currents allows the direct implementation of the synthesised real antenna [46].

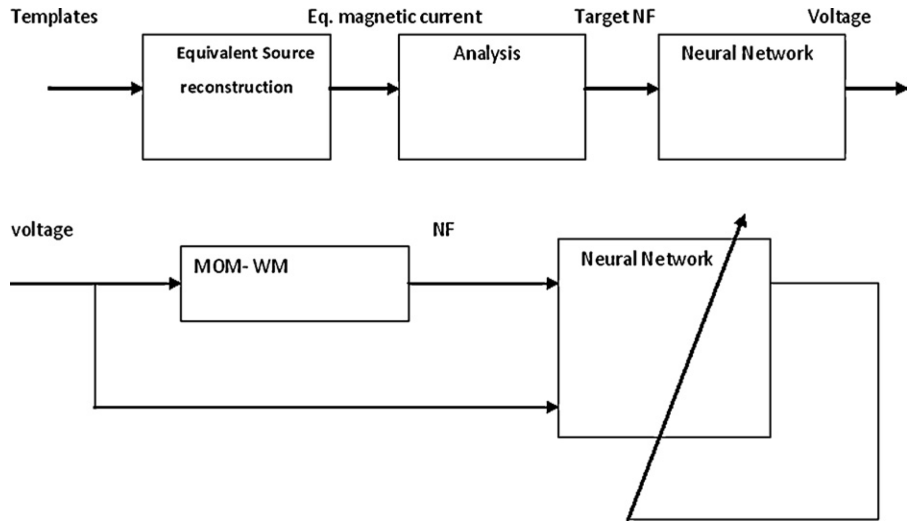

Fig. 5. Training procedure.

6.2. Array synthesis using ANN and GA

The NN and GA can be combined in a new hybrid technique that can perform synthesis tasks over passive element antenna arrays, whose radiation pattern is defined by the coupling effects between the active and passive elements. The combination of both methods improves the results obtained using NN, also reducing the computation time required by GA. Coupled ANN and GA models are developed for the design of broadband tapered slot antenna array elements [49]. The ANN is employed to establish the complicated relationships between the key array performance indicator, i.e., active reflection coefficient, and its element parameters. The trained ANN models are combined with the GA to optimise the element parameters for a given operating frequency band without using time-consuming electromagnetic (EM) simulators. The optimisation results show that the developed ANN-GA model can retain the accuracy of EM simulators and exhibit high computational efficiency [47].

6.3. Array pattern synthesis with mutual coupling effect using wavelet neural networks

The structure of an ANN is an array of processing neurons ordered in layers with a network of interconnection weights, w_j ($j=1,2,\ldots L$) between the neurons in different layers. The input data x_j are processed through an activation function $f(x)$. The output y of the neuron in this model is given by

$$y = f\left(\sum\nolimits_{j=1}^{L} w_j x_j + b\right) \tag{19}$$

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 $w_j=b$, the bias can be calculated in the training phase. The inputs to the network are samples of the desired pattern. The number of input neurons is dependent on the accuracy needed to represent the desired pattern. The outputs of the network are the amplitude and phase distribution values of the array elements. The specific NN algorithm with one hidden layer, based on wavelet activation functions and trained with examples generated by the Orchard–Elliott method, is presented in [54]. This algorithm accounts for the mutual coupling effect in a linear dipole array with a shaped pattern. The excitations and the dipole lengths in the array are determined. The MSE criterion is used to quantify the closeness between the desired and computed patterns [51]. The NN architecture considered in this approach has one hidden layer and uses wavelet functions as activation functions as described in [55]. Owing to the special characteristics of the wavelet functions, wavelet neural networks (WNN) can better approximate patterns with sharp transitions [55]. The chosen wavelet activation function $f(y_i)$ is the inverse Mexican hat function [56] given by

$$f(y_i) = (y_i^2 - 1)\exp\left(-\frac{y_i^2}{2}\right) \tag{20}$$

where

$$y_i = \left|\frac{x - t_i}{a_i}\right| = \sqrt{\sum_{j=1}^{L}\left(\frac{x_j - t_{ij}}{a_i}\right)^2}$$

7. Conclusions

In this paper, NN applications to SAA are reviewed briefly. The NN method is more accurate and faster than conventional linear-algebra-based methods because NN can consider real-time properties, whereas conventional methods use the array factor mechanism only. Most of the applications of smart antenna arrays can be solved by the MLPNN, RBFNN or HNN models, but a hybrid method or other neural model (WNN) can also be used for certain cases. The MLPNN is suitable for fault-finding, DOA estimation and null enforcement; RBFNN is suitable for obstacle modelling, DOA estimation, digital beamforming and finding defective elements; and HNN is suitable for tracking and adaptive beamforming. Hybrid techniques (source reconstruction or GA) and WNN can be used where coupling is significant. Table 4 compares all available NN applications in antenna array. The RBFNN method is much faster than the MLPNN method and suitable for approximating nonlinear mapping applications, such as finding the weights for an antenna array, direction and fault-finding. The HNN method can perform energy minimisation to solve optimisation problems and is suitable for tracking and adaptive beam forming purposes. MLPNN provides excellent result for null enforcement. Much work remains to further improve the design of smart antennas, especially for real-time application such as mobile and satellite communications.

Table 4
Comparison of different applications of NN in antenna array.

Type of neural network	Fault finding	DOA estimation	Obstacle modelling	Beamforming	Tracking	Real-time array synthesis	Null enforcement in field pattern
MLP	Locate fault element in space platform [15], locate fault element in linear array [17], find fault element [18] with backpropagation mode in all cases	DOA application in linear array with feed-forward mode [28]	NA	NA	NA	NA	Null enforcement of array installed on geostationary satellite using hyperbolic tangent sigmoid transfer function with resilient backpropagation algorithm [45]
RBF	Locate fault element with backpropagation mode [17]	Detection and DOA with N-MUST algorithm [28], direction finding with trains using a least mean squared (LMS) error solution [29]	Taking consideration of conducting obstacle with *k*-means algorithm [26]		Change the weight of 1d and 2d array for fast tracking [27]	NA	NA
HNN	NA	NA	NA	Adaptive beam forming with direct sequence spread spectrum, single-layer feed-forward [37]	Neural tracker with phased array antenna, single-layer feed-forward [43]	NA	NA
Hybrid NN	NA	NA	NA	NA	NA	Source reconstruction considering real-time coupling effect [46], Tapered slot antenna array with coupling effect [49]	NA
Wavelet NN	NA	NA	NA	NA	NA	Linear dipole array synthesis with shaped pattern [51]	NA

References

[1] Ueng Fang-Biau, Chen Jun-Da, Cheng Sheng-Han. Smart antennas for multi-user DS/CDMA communications in multipath fading channels. In: IEEE eighth international symposium on spread spectrum. 2004. p. 400–4.
[2] Alexiou A, Haardt M. Smart antenna technologies for future wireless systems: trends and challenges. IEEE Commun Mag 2004;42(September):90–7.
[3] Rawat A, Yadav RN, Shrivastava SC. Neural modelling of 15 element dynamic phased array smart antenna. In: IEEE international conference on advances in computing, control, and telecommunication technologies. 2009. p. 45–9.
[4] Boukalov AO, Hiiggman SG. System aspects of smart-antenna technology in cellular wireless communications—an overview. IEEE Trans Microw Theory Tech 2000;6:919–28.
[5] Chryssomallis M. Smart antennas. IEEE Antenna Propagation Mag 2000:129–36.
[6] Zhang Jian-Wu. The adaptive algorithms of the smart antenna system in future mobile telecommunication systems. IEEE Int Workshop Antenna Technol 2005:347–50.
[7] Lehne PH, Pettersen M, Telenor. An overview of smart antenna technology for mobile communications systems. IEEE Commun Surv 1999.
[8] Bang S, Sheu SH, Bing J. Neural network for detection of signals in communication. IEEE Trans Circ Syst 1996;43:644–55.
[9] Lim CP, Harrison RF. Online pattern classification with multiple neural network systems: an experimental study. IEEE Trans Syst, Man, Cybern 2003;33: 235–47.
[10] Widrow B, Lehr M. A 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proc IEEE 1990;78(September):1415–42.
[11] Nair SK, Moon J. A theoretical study of linear and nonlinear equalization in the nonlinear magnetic storage channels. IEEE Trans Neural Netw 1997;8(September):1106–18.
[12] Christodoulou C, Georgiopoulos M. Applications of neural networks in electromagnetics. Boston: Artech House Inc.; 2001.
[13] Al-Bajari M, Ahmed JM, Ayoob MB. Performance evaluation of an artificial neural network-based adaptive antenna array system. Lecture Notes Comput Sci 2010;Issue Part 2(6025):11–20.
[14] Yeo BK, Lu Y. Array failure correction with a genetic algorithm. IEEE Trans Antenna Propagation 1999;47:823–8.
[15] Patnaik A, Choudhury B, Pradhan P, Mishra RK, Christodoulou C. An ANN application for fault finding in antenna arrays. IEEE Trans Antenna Propagation 2007;55(March):775–7.
[16] Nagatsuka M, Ishii N, Kohno R, Imai H. Adaptive array antenna based on spatial spectral estimation using maximum entropy method. IEICE Trans Commun 1994;11:624–33.
[17] Rodriguez JA, Fernandez-Delgado M, Bregains J. A comparison among several techniques for finding defective elements in antenna array. In: EUCAP the second European conference on antenna and propagation. 2007. p. 1–8.
[18] Acharya OP, Patnaik A, Choudhury B. A PSO application for locating defective elements in antenna arrays. World Congr Nat Biological Inspired Comput 2009:1094–8.
[19] Patnaik A, Christodoulou C. Finding failed element positions in linear antenna arrays using neural networks. IEEE Antenna Propagation Soc Symp 2006:1675–8.
[20] Rodriguez JA, Ares F, Lepez P, Moreno E. A simple way of obtaining optimized patterns using the Woodward–Lawson method. IEEE Antenna Propagation Mag 2006;48:100–3.
[21] Hunter A. SUGAL genetic algorithm package v. 2.1. England: University of Sunderland; 1995.
[22] Rodriguez JA, Ares F, Palacios H, Vassal'lo J. Finding defective elements in planar arrays using genetic algorithms. In: Progress in electromagnetics research, PIER 29. 2000. p. 25–37.
[23] Haykins S. Neural networks: A comprehensive foundation. New York: IEEE Press/IEEE Computer Society Press; 1994.
[24] Rodriguez JA, Ares F, Moreno E. Waveguide-fed longitudinal slot array antennas: fault diagnosis using measurements of input impedance. Microw Opt Technol Lett 2002;32:200–1.
[25] Aumann HM, Fenn AJ, Willwerth FG. Phased array antenna calibration and pattern prediction using mutual coupling measurements. IEEE Trans Antenna Propagation 1989;37:844–50.
[26] Ayestarán RG, Las-Heras F. Obstacle modelling in array synthesis using neural networks. IEEE Trans Antenna Propagation 2006;54:2420–4.
[27] Zooghby HEl, Christodoulou CG, Georgiopoulos M. Neural network-based adaptive beamforming for one- and two-dimensional antenna arrays. IEEE Trans Antenna Propagation 1998;46:1891–3.
[28] Toh Kar-Ann, Lee Chong-Yee. Efficient network training for DOA estimation. In: Conference proceeding of systems, man, and cybernetics, vol. 5. 1999. p. 383–8.
[29] Ahmed H, El Zooghby, Christos G, Christodoulou, Georgiopoulos Michael. A neural network-based smart antenna for multiple source tracking. IEEE Trans Antenna Propagation 2000;48:768–75.

[30] Zooghby Ahmed HEl, Christodoulou Christos G, Georgiopoulos M. Performance of radial-basis function networks for direction of arrival estimation with antenna arrays. IEEE Trans Antenna Propagation 1997;45:1611–7.
[31] Chen Sun, Nemai C, Karmakar. Direction of arrival estimation with a novel single-port smart antenna. EURASIP J Appl Signal Process 2004;9:1364–75.
[32] Zooghby AHEL, Southall HL, Christodoulou CG. Experimental validation of a neural network direction finder. IEEE Antenna Propagation Soc Int Symp 1999:1592–695.
[33] Schmidt RO. Multiple target location and signal parameter estimation. IEEE Trans Antenna Propagation 1986 Mar;AP-34(3):276–80.
[34] Hsueh-Jyh Li, Rong-Yuan Lane. Utilization of multiple polarized data for aerospace target identification. IEEE Trans Antenna Propagation 1995;43:1436–40.
[35] Southall Hugh L, Jeffrey A, Simmers, Teresa H O'Donnell. Direction finding in phased arrays with a neural network beamformer. IEEE Trans Antenna Propagation 1995;43:1369–74.
[36] Quach B, Leung H, Lo T, Litva J. Hopfield network approach to beamforming in spread spectrum communication. In: IEEE proceedings of the seventh SP workshop on statistical signal & array processing. 1994. p. 409–12.
[37] Sandhu, Titus Lo, Leung H, Litva J. Hopfield neurobeam former for spread spectrum communications. In: 6th IEEE international symposium on PIMRC. 1995 Sep. p. 926–30.
[38] Akyildiz A. Survey on sensor networks. IEEE Commun Mag 2002;40(August):102–14.
[39] Sun Q, Alouani AT, Rice TR, Gray JE. Linear system state estimation: a neurocomputing approach. In: Proceedings of the American control conference. 1992. p. 550–4.
[40] Dilsavor Ronald L, Moses Randolph L. Analysis of modified SMI method for adaptive array weight control. IEEE Trans Signal Process 1993;41:721–6.
[41] Alamouti SM. A simple transmit diversity technique for wireless communications. IEEE J Selected Areas Commun 1998;16:1451–8.
[42] Tarokh V, Jafarkhani H, Calderbank AR. Space–time block codes from orthogonal designs. IEEE Trans Inform Theory 1999;45:1456–67.
[43] Leung H. A Hopfield neural tracker for phased array antenna. IEEE Trans Aerospace Electron Syst 1997;33:301–7.
[44] Hansen RC. Phased array antennas. New York: Wiley; 1998.
[45] Brégains JC, Dorado J, Gestal M, Rodríguez JA, Ares F, Pazos A. Avoiding interference in planar arrays through the use of artificial neural networks. IEEE Antenna Propagation Mag 2002;44:61–5.
[46] Ayestarán RG, Las Heras F. Neural networks and equivalent source reconstruction for real antenna array synthesis. IEE Electron Lett 2003;39:956–8.
[47] Wang S, Chen X, Parini CG, McCormick J. Design of broadband array elements based on neural network and genetic algorithm. Electron Lett 2009;45: 1011–2.
[48] Rafael G Ayestarán, Las Heras Fernando, Luis F Herrán. Neural modelling of mutual coupling for antenna array synthesis. IEEE Trans Antenna Propagation 2007 Mar;55:832–40.
[49] Ayestarán RG, Las Heras F, Martínez José A. Non-uniform antenna array synthesis using neural networks. J Electromagn Waves Appl 2007;21: 1001–11.
[50] El Zooghby, Ahmed. Smart antenna engineering. Artech House; 2005.
[51] Shavit R, Taig I. Array pattern synthesis using neural networks with mutual coupling effect. IEE Proc Microw, Antenna Propagation 2005;152:354–8.
[52] Janaswamy Ramakrishna. Radio wave propagation and smart antennas for wireless communications. Kluwer Academic; 2002.
[53] Gross Frank. Smart antennas for wireless communications with MATLAB. McGraw-Hill; 2005.
[54] Orchard HJ, Elliott RS, Stern GJ. Optimising the synthesis of shaped antenna patterns. IEE Proc Microw Opt Antenna 1985;132:63–8.
[55] Shavit R, Taig I. Comparison study of pattern synthesis techniques using neural networks. Microw Opt Technol Lett 2004;42:175–9.
[56] Zhang Q. Wavelet networks: The radial structure and an efficient initialization procedure. Technical Report LiTH–ISY–I–1423, Linkö ping University, 1992.
[57] Balanis CA. Antenna theory. New York: Wiley; 1981.
[58] Heras F Las, Sarkar TK. A direct optimization approach for source reconstruction and NF-FF transformation using amplitude only data. IEEE Trans Antenna Propagation 2002;50:500–10.
[59] Rivas A, Rodríguez JA, Ares F, Moreno E. Planar arrays with square lattices and circular uniform amplitude or very low dynamic range ratio. IEEE Antenna Propagation Mag 2001;43:90–3.
[60] Ares F, Rodríguez JA, Villanueva E, Rengajaran SR. Genetic algorithms in the design and optimization of antenna array pattern. IEEE Trans Antenna Propagation 1999;47:506–10.
[61] Ares F, Rengajaran SR, Villanueva E, Skochinski E, Moreno E. Application of genetic algorithms and simulated annealing technique in optimizing the aperture distributions of antenna array patterns. Electron Lett 1996;32:148–9.

Abhishek Rawat received his B.E. Degree in Electronics and Communication Engineering and M.Tech. in Microwave Engineering from R.G.P.V. and Maulana Azad National Institute of Technology, Bhopal, India in 2001 and 2006 respectively. He is pursuing Ph.D. from Maulana Azad National Institute of Technology, Bhopal, India. He is presently working as assistant professor in Samrat Ashok Technological Institute, Vidisha, India. His main interest is Antenna array synthesis and design, optimization techniques in electromagnetic. He is also received young scientist award in 2007 by the Madhya Pradesh Council of Science and Technology, Bhopal. He is life Associate Member of IETE and authored about 10 papers in International Journals and Conferences.

R.N. Yadav received his B.E. Degree from Motilal Nehru Regional Engineering College (Now MNNIT) Allahabad, M.Tech. degree from Maulana Azad College of Technology (Now MANIT) Bhopal and Ph.D. Degree from Indian Institute of Technology, Kanpur, India in 1993, 1997 and 2005 respectively. Currently he is Associate Professor with Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, India. His area of interest Neural Networks applications in Electronics and Communication engineering. He has authored about 50 papers in International Journals and Conferences. He is the life member of IETE and ICEIT.

S.C. Shrivastava received his Bachelor and Master degree in Engineering from Government Engineering College, Jabalpur, India in 1968 and 1970 respectively and PhD. degree in 1994 from Barkatullah University, Bhopal, India. He is a life member of IETE, IE(I) and ISTEE. He was Head of Electronics and Communication Engineering department, Maulana Azad National Institute of Technology, Bhopal, India. Currently he is Director (PG), T.I.T. Institute of Science and Technology, Bhopal, India.