A Neural Network-Based Smart Antenna for Multiple Source Tracking

Ahmed H. El Zooghby, *Student Member, IEEE*, Christos G. Christodoulou, *Senior Member, IEEE*, and Michael Georgiopoulos

***Abstract*—This paper considers the problem of *multiple-source* tracking with neural network-based smart antennas for wireless terrestrial and satellite mobile communications. The neural multiple-source tracking (N-MUST) algorithm is based on an architecture of a family of radial basis function neural networks (RBFNN) to perform both detection and direction of arrival (DOA) estimation. The field of view of the antenna array is divided into spatial angular sectors, which are in turn assigned to a different pair of RBFNN's. When a network detects one or more sources in the first stage, the corresponding second stage network(s) are activated to perform the DOA estimation. Simulation results are performed to investigate the performance of the algorithm for various angular separations, with sources of random relative signal-to-noise ratio and when the system suffers from a doppler spread.**

***Index Terms*—Direction-of-arrival estimation, multibeam antennas, neural networks.**

I. Introduction

THE concept of frequency reuse has been successfully implemented in modern cellular communications systems in order to increase the system capacity. Extensive research has showed that further improvement can be achieved by employing adaptive arrays at the base station [1], [2]. In order to accomplish tracking of desired users, direction finding algorithms are used to locate the positions of the mobile users as they move within or between cells. On the other hand, as a growing number of mobile satellite communication systems are being introduced and as global positioning systems (GPS) systems become more widely used, smart antennas capable of separating signals from multiple sources can substantially improve the performance of those systems as well. Hence, a direction finding algorithm that can operate in real time is an integral part of any spatial division multiple access (SDMA) scheme for terrestrial as well as satellite mobile communication systems. Other applications of direction finding include target tracking and telemetry.

Superresolution algorithms [3] have been successfully applied to the problem of direction-of-arrival (DOA) estimation to locate radiating sources with additive noise, uncorrelated, and correlated signals. One of the main disadvantages of the superresolution algorithms is that they require extensive computation and as a result they are difficult to implement in real time. Recently, neural networks-based direction finding algorithms have been proposed for single and multiple source direction finding ([4]–[6]). It has been shown that neural networks have the capability to track sources in real time. In [7], a radial basis function neural network has been used to track the locations of mobile users. However, a different network had to be used for different number of users with some fixed angular separation. This paper presents a generalization of the algorithm introduced in [7] in such a way that the system would be able to track an arbitrary number of sources with any angular separation *without prior knowledge* of the number of sources. The neural multiple source tracking (N-MUST) algorithm is based on an architecture of a family of radial basis function neural networks that perform both detection and DOA estimation. The new approach is based on dividing the field of view of the antenna array into angular spatial sectors, then train 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 so as to estimate the exact location of the sources.

The main advantage of this new approach is a dramatic reduction in the size of the training set required to train each smaller neural network. Results for the DOA estimation of multiple sources using this new approach are presented and discussed. The organization of this paper is as follows: Section II presents the problem formulation and elaborates on the use of neural networks for direction finding. In Section III, the new approach labeled N-MUST is detailed as well as the different detection and DOA estimation steps. The simulations results are presented in Section IV and in Section V some conclusive remarks summarize the performance of the algorithm.

II. Neural Network-Based Direction Finding

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

$$x_i(t) = \sum_{m=1}^{K} s_m(t) e^{-j(i-1)k_m} + n_i(t) \quad ; i = 1, 2, \cdots M \tag{1}$$

Manuscript received Septmber 29, 1998; revised November 15, 1999.

A. H. El Zooghby is with Qualcomm Global Services, San Diego, CA 92191 USA.

C. G. Christodoulou is with the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, NM. 87131 USA.

M. Georgiopoulos is with the Electrical and Computer Engineering Department, University of Central Florida, Orlando, FL 32816 USA.

Publisher Item Identifier S 0018-926X(00)04379-9.

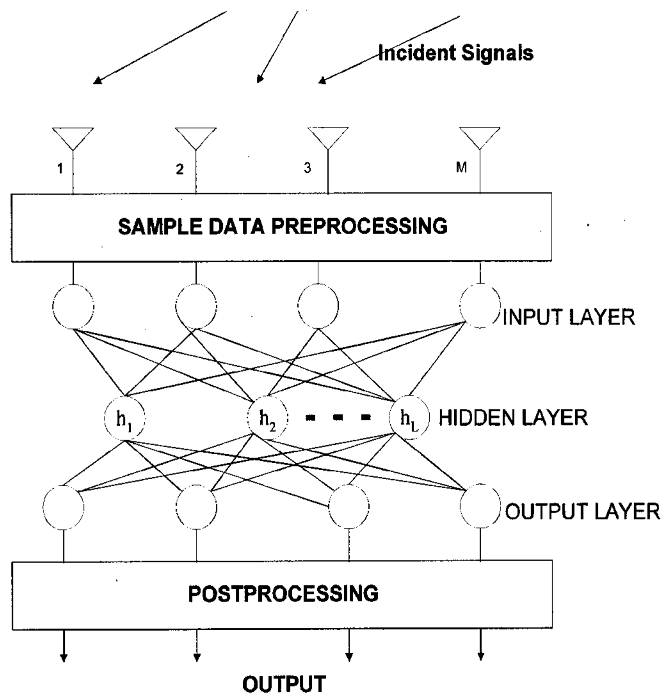


Fig. 1. The block diagram of an RBFNN with pre- and postprocessing stages.

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

$$k_m = \frac{\omega_0\, d}{c}\sin(\theta_m) \tag{2}$$

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 a matrix form

$$\boldsymbol{X}(t) = \boldsymbol{A}\boldsymbol{S}(t) + \boldsymbol{N}(t) \tag{3}$$

where $\boldsymbol{X}(t)$, $\boldsymbol{N}(t)$, and $\boldsymbol{S}(t)$ are given by

$$\boldsymbol{X}(t) = [\,x_1(t) \quad x_2(t) \quad \cdots \quad x_M(t)\,]^T \tag{4}$$

$$\boldsymbol{N}(t) = [\,n_1(t) \quad n_2(t) \quad \cdots \quad n_M(t)\,]^T \tag{5}$$

$$\boldsymbol{S}(t) = [\,s_1(t) \quad s_2(t) \quad \cdots \quad s_K(t)\,]^T. \tag{6}$$

In (4)–(6) the superscript "T" indicates the transpose of the matrix. Also in (3) $\boldsymbol{A}$ is the $M \times K$ steering matrix of the array toward the direction of the incoming signals defined as

$$\boldsymbol{A} = [\,a(\theta_1) \quad \cdots \quad a(\theta_m) \quad \cdots \quad a(\theta_K)\,] \tag{7}$$

where $\boldsymbol{a}(\theta_m)$ corresponds to

$$\boldsymbol{a}(\theta_m) = [\,1 \quad e^{-jk_m} \quad e^{-j2k_m} \quad \cdots \quad e^{-j(M-1)k_m}\,]. \tag{8}$$

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 also independent of $\boldsymbol{S}(t)$, then the received spatial correlation matrix $\boldsymbol{R}$ of the received noisy signals can be expressed as

$$\begin{aligned}\boldsymbol{R} &= E\{\boldsymbol{X}(t)\boldsymbol{X}(t)^H\} \\ &= \boldsymbol{A}E[\boldsymbol{S}(t)\boldsymbol{S}^H(t)]\boldsymbol{A}^H + E[\boldsymbol{N}(t)\boldsymbol{N}^H(t)].\end{aligned} \tag{9}$$

In the above equation, "H" denotes the conjugate transpose. The antenna array can be thought of as performing a mapping G: $\boldsymbol{R}^K \rightarrow \boldsymbol{C}^M$ from the space of the DOA's, $\{\Theta = [\theta_1, \theta_2, \cdots, \theta_K]^T\}$ to the space of sensor output $\{\boldsymbol{X}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_M(t)]^T\}$. A neural network is used to perform the inverse mapping F: $C^M \rightarrow R^K$. The algorithm described in this paper for the problem of direction finding is based on using radial basis function neural networks to approximate this inverse mapping F. Note that an RBFNN can approximate an arbitrary function from an input space of arbitrary dimensionality to an output space of arbitrary dimensionality ([8]–[10]). The reason for choosing a RBFNN is because it trains faster than its backprop NN counterpart. The block diagram of an RBFNN is shown in Fig. 1. In between the blocks designated "sample data processing" and "postprocessing," as can be seen from Fig. 1, the RBFNN consists of three layers of nodes: the input layer, the output layer, and the hidden layer. The input layer is the layer where the inputs are applied; the output layer is the layer where the outputs are produced. As is the case with most neural networks, the RBFNN is designed to perform an input/output mapping trained with examples. The purpose of the hidden layer in an RBFNN is to transform input data from an input space of some dimensionality to a new space of possibly higher dimensionality (see Fig. 1). The rationale behind this transformation is based on Cover's theorem [11], which states that an input/output mapping problem cast in a high-dimensionality space nonlinearly is easier to solve. The nonlinear functions (the h's in Fig. 1) that perform this transformation are usually taken to be Gaussian functions of appropriately chosen means and variances. There are a lot of learning strategies that have appeared in the literature to train a RBFNN. The one used in this paper was introduced in [12], where an unsupervised learning algorithm (such as the K-means [13]) is initially used to identify the centers of the Gaussian functions used in the hidden layer. Then, an *ad-hoc* procedure is used to determine the widths (standard deviations) of these Gaussian functions. According to this procedure, the standard deviation of a Gaussian function of a certain mean is the average distance to the first few nearest neighbors of the means of the other Gaussian functions. The aforementioned unsupervised learning procedure allows you to identify the weights (means and standard deviations of the Gaussian functions) from the input layer to the hidden layer. The weights from the hidden layer to the output layer are identified by following a supervised learning procedure, applied to a single layer network (the network from hidden to output layer). This supervised rule is referred to as the delta rule. The delta rule is essentially a gradient decent procedure applied to an appropriately defined optimization

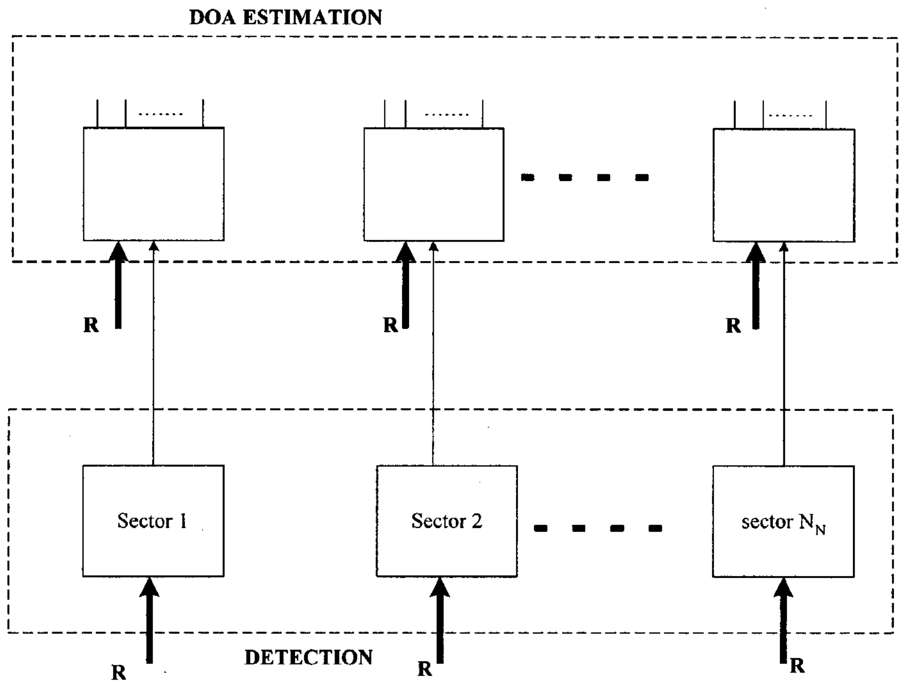


Fig. 2. The neural multiple source tracking architecture.

problem. For more details about the *delta rule* and how it is applied to single layer networks, see [9].

It should be mentioned here that although we could have used the n-foldout technique for training, our training approach proved to be less time consuming and yielded fairly accurate results. In the n-foldout technique the training set is split into n subsets. Then an RBFFN is trained on all subsets except for one and the error is measured on the subset left out. This procedure is repeated for a total of n times, each time using a different subset for error measurement and the remaining subsets for training. Although, this statistical technique of measuring the error is more reliable than our procedure, it is extremely time consuming . Since, in our problem, the data can be of large size the n-fold holdout technique can be even more time consuming. This is the main reason why we have avoided the n-fold holdout technique.

Once training of the RBFNN is accomplished, the training phase is complete, and the trained neural network can operate in the performance mode (phase). In the *performance (testing) phase*, the neural network is expected to generalize, that is respond to inputs that it has never seen before, but drawn from the same distribution as the inputs used in the training set. One way of explaining the generalization exhibited by the network during the performance phase is by remembering that after the training phase is complete the RBFNN has established an approximation of the desired input/output mapping. Hence, during the performance phase the RBFNN produces outputs to previously unseen inputs by interpolating between the inputs used (seen) in the training phase.

A. Sample Data Preprocessing

In general, array processing algorithms utilize the correlation matrix for direction of arrival estimation purposes instead of the actual array output $\boldsymbol{X}(t)$ since it contains sufficient information about the received signals. The input vector to the input layer of the network (see Fig. 1) is the spatial correlation matrix $\boldsymbol{R}$ that can be organized as an M^2-dimensional vector denoted by $\boldsymbol{b}$. It then follows that the number of input units at the input layer is given by $2M^2$. This is due to the fact that we need twice as many input nodes for the neural network since the network does not deal directly with complex numbers. The dimension of the hidden layer is equal to the number of the Gaussian functions L, which can be chosen to be equal to the number of total input/output pairs in the training set if perfect recall is desired. The input vector b is normalized by its norm prior to being applied at the input layer of the neural network, i.e.,

$$\boldsymbol{z} = \frac{\boldsymbol{b}}{||\boldsymbol{b}||}. \tag{10}$$

In order to reduce the dimension of the input layer, other preprocessing schemes have been suggested by researchers. The sum of the diagonal of the correlation matrix helps reduce the number of input nodes needed to $2M$ for an M element linear array. However, this comes with the price of reduced network generalization performance. By exploiting the symmetry in the correlation matrix R one need only consider either the upper or lower triangular part of the matrix. In our design, the upper triangular half of $\boldsymbol{R}$ is used. An $M \times M$ spatial correlation matrix $\boldsymbol{R}$ can organized in an $M(M+1)$-dimensional vector of real and imaginary parts denoted $\boldsymbol{b}$. This procedure is illustrated in the following:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$\boldsymbol{b} = [r_{11} \quad r_{12} \quad r_{13} \quad r_{22} \quad r_{23} \quad r_{33}].$$

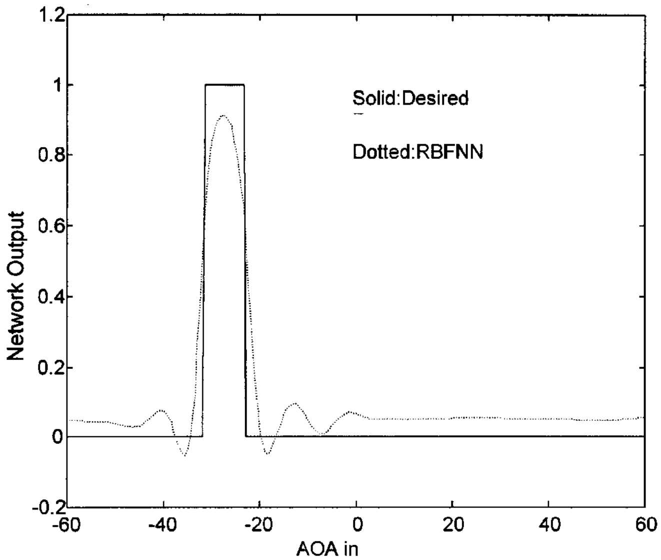


Fig. 3. A ten-element linear array and sources in a 10° wide sector [−25° −35°]. Trained with two sources and tested with four sources separated by 2° in space.

It should be noted here that training a single neural network to detect the angle of arrival of multiple sources is not an easy task. To get an idea of how much training is required, consider the problem of tracking two sources only. First, we start with training the network with a 2° angular separation from −90° to 90°. That means you set the first source at $\theta = -90°$ and the next source at $\theta = -88°$, next you set the first source at −89° and the second to −87°, and so on until you cover the region of interest (assuming −90° to 90°). Next you repeat the same training procedure with 3° difference between the two sources (−90 and −87, −89 and −86, −88 and −85 $\cdots$ 87, and 90). Then 5° (−90 and −85, −89, and −84, −88 and −83, etc.), then 10 degrees apart: 15, 20, 30, 40, 50, and 60. Thus, the training set consists of sources with angle of arrivals that cover a wide range possibilities. The testing is done by presenting to the network data corresponding to two sources with angles of separation that it has not seen before. For example, if two sources that are separated by 12° apart are presented, the network can detect these two sources accurately by quickly interpolating between the responses it was trained to produce during the training phase. This kind of exhaustive training becomes prohibitive for more than three or four sources since the number of possible training data combinations is enormous. To circumvent this problem, multiple, but smaller, neural networks are employed. Each network then tracks a smaller number of sources within a smaller angular sector.

III. Neural Multiple Source Tracking (N-MUST) Algorithm

The N-MUST algorithm is also based on the RBFNN, but it is composed of two stages—the detection stage and the *estimation* stage—as shown in Fig. 2. In the first stage, a number of RBFNN's are trained to perform the detection phase, while in the second stage another set of networks is trained for the direction of arrival estimation phase. When networks detect one or more sources in the first stage, the corresponding second stage networks are activated to perform the direction of arrival (DOA) estimation step. No prior knowledge of the number of present sources is required.

A. Detection Stage

In this approach, labeled the N-MUST algorithm, an arbitrary number of mobile users (sources) can be tracked and no prior knowledge of the number of mobile users is required. As shown in Fig. 2, there are two stages of RBFNN's. The first stage is the "detection stage," which consists of P RBFNN's, each of width θ_W. The entire angular spectrum (field of view of the antenna array) is divided in P sectors. The pth $(1 \leq p \leq P)$ RBFNN is trained to determine if one or more signals exist within the $[(p-1)\theta_W, p\theta_W)]$ sector. If there are any signals present in the corresponding sector, the neural network will give the value one for an answer. Otherwise, the network will register a zero as its output value. This information is then passed to the second stage, the "direction of arrival" stage, which estimates the angles of these signals.

Each one of the P neural networks of the detection stage, has $M(M+1)$ input nodes representing the correlation matrix $\boldsymbol{R}$ and one output node. The number of hidden nodes in the second layer is also $M(M+1)$. The training procedure for a network in the detection stage is outlined below.

1) Network Training in the Detection Stage:

1) Evaluate the correlation matrix of the nth array output vector using equation (9) $\{\boldsymbol{R^n}, n = 1, 2, \cdots, N\}$.
2) Form the vectors $\{\boldsymbol{b^n}, n = 1, 2, \cdots, N\}$.
3) Normalize the input vectors using equation (10).
4) Generate input output pairs $\{\boldsymbol{z^n}, 1\}$ for sources located in the sector, and $\{\boldsymbol{z^n}, 0\}$ for sources located outside the sector where $n = 1, 2, \cdots, N$.
5) Employ an appropriate RBFNN in the detection to learn the training set generated in step 4).

2) Test (Generalization) Phase in the Detection Stage:

1) Evaluate the sample correlation matrix using the collected array output measurements using equation (9).
2) Form the vectors $\hat{\boldsymbol{b}}$.
3) Produce the normalized input vectors $\hat{\boldsymbol{z}}$ using equation (10).
4) Present input vectors $\hat{\boldsymbol{z}}$ to the RBFNN's of the detection stage and obtain an output $\{0 \text{ or } 1\}$ from each one.

To illustrate how a network is trained in the detection stage, let us consider a case where the network is required to track N_s sources in the [10° 20°] sector with some angular separation $\Delta\theta$. We start the training with sources at $-90°, -90° + \Delta\theta, \cdots, -90° + (N_s - 1)\Delta\theta$. We use this vector of DOA to generate the correlation matrix $\boldsymbol{R}$ and the normalized vector $\boldsymbol{z}$. Since the sources are outside the sector of interest, the target output is "0" in this case. We then select the subsequent DOA vectors as $-88°, -88° + \Delta\theta, \cdots, -88° + (N_s - 1)\Delta\theta, -86°, -86° + \Delta\theta, \cdots, -86° + (N_s - 1)\Delta\theta$ and so on. The target output of the network is set to "1" only when one or more of the angles in the DOA vector lies in the

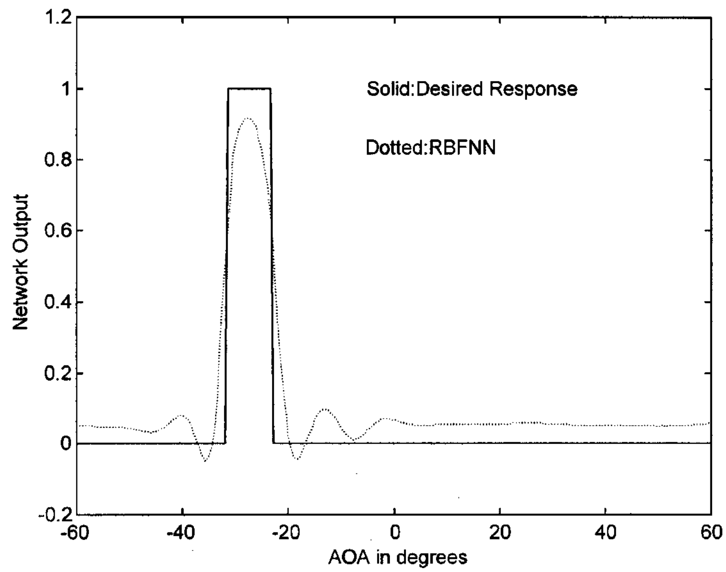


Fig. 4. Comparison between the desired and actual response of a ten-element array trained with two equipower sources of 2° angular separation and tested with three sources of 3° angular separation and different SNR's.

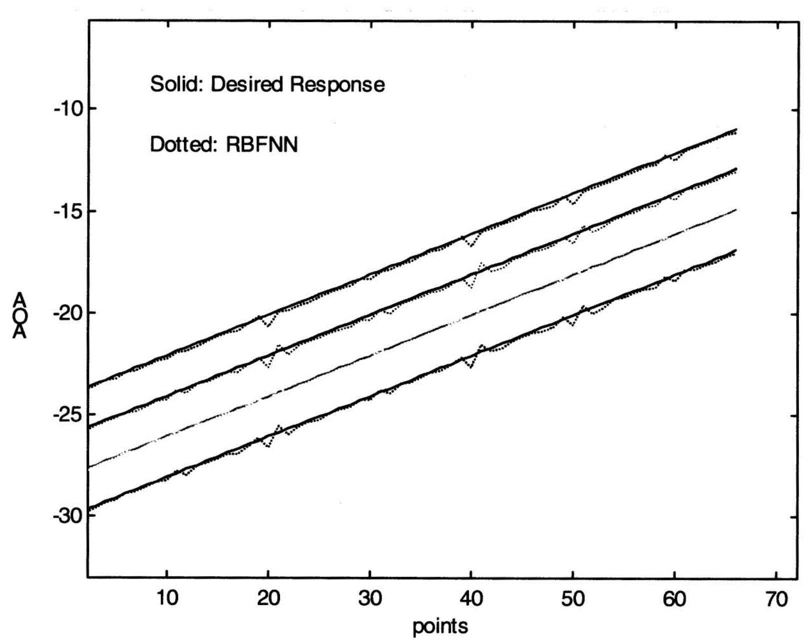


Fig. 5. Response of an eight-element linear array $(d = \lambda/2)$ tracking four sources of 2° angular separation in the sector [−30° −11°]. The sources are of equal power, 5 dB higher than the noise power.

[10° 20°] range. In the simulations performed, a network was tested with number of sources and angular separations different than it had seen in the training. The network was able to detect the presence of the sources correctly. This suggests that considering all possible combinations of number of sources and separations need not be considered for the detection phase.

B. DOA Estimation Stage

The second stage of neural networks is trained to perform the actual direction of arrival estimation. The P networks of the DOA estimation stage are assigned to the same spatial sectors as in the detection stage (see Fig. 2). When the output of one or more networks from the first stage is 1, the corresponding second stage network(s) are activated. The input information to each second stage network is the correlation matrix $\boldsymbol{R}$, while the output is the actual DOA of the sources. The number of hidden nodes is the same as the number of input nodes given by $M(M+1)$. The optimum size of hidden nodes is not always easy to determine. In this work, after extensive experimentation, it was found that by choosing the number of hidden nodes to be equal to or larger than the number of input nodes, good results can be obtained.

Consider a system with minimum source resolution of 2°, a single neural network trained to track sources over the antenna's

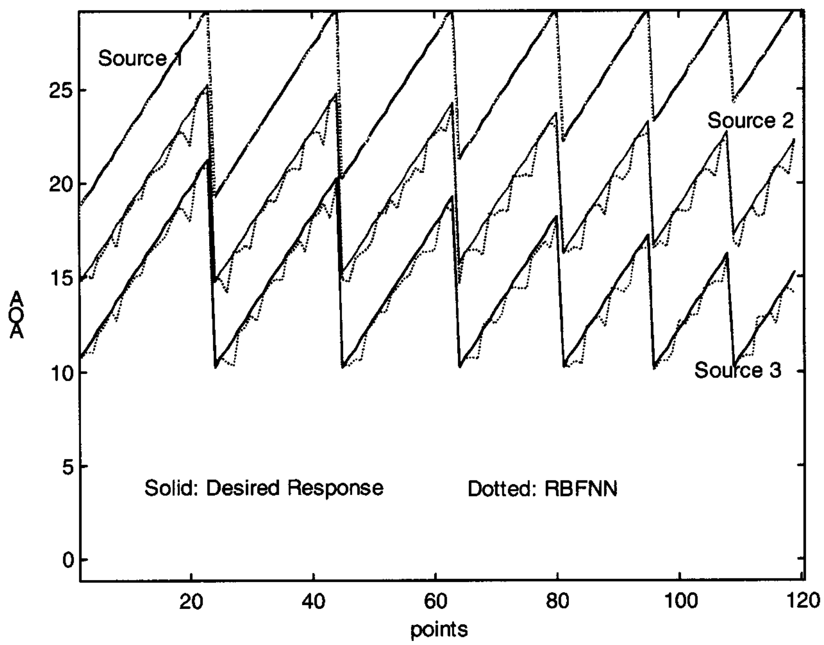


Fig. 6. Response of a 12-element array which was trained with d/λ ranging from 0.4–0.6 and with three sources 4°, 4.5°, 5°, $\cdots$, 7° of angular separation in the sector [10° 29°].

field of view (e.g., $180°$ wide) could be trained for angular separations $\Delta\theta$ of $2°$, $4°$, $6°$, $\cdots$ up to some $\Delta\theta$. This results in such a huge training set that the single neural network approach becomes impractical. However, by assigning different networks for different angular spatial sectors, smaller training sets are sufficient since the network is only required to track sources in a limited spatial region. For sectors 10°, $-20°$ wide, it follows that the number of distinct locations of possible sources as well as the size of the training set are significantly reduced. Whereas most direction finding algorithms require the knowledge of the number of sources, in our approach we only need to specify the minimum angular resolution that the system is required to achieve. Rather than designing the network with number of output nodes equal to K (number of sources), for a sector of width θ_W and minimum angular resolution of $\Delta\theta_{\min}$, the number of output nodes is given by

$$J = \left\lfloor \frac{\theta_W}{\Delta\theta_{\min}} \right\rfloor. \tag{11}$$

DOA estimates are obtained by postprocessing the neural network outputs of the second stage. J output nodes represent bins in a discrete angular spatial region centered at $\Delta\theta_{\min}$ intervals. The output nodes are trained to produce values between "0" and "1." An output of "1"indicates the presence of a source exactly on the bin and a "0" represents no source. Sources located between the bin angles are represented by values between "0" and "1."

1) Network Training Phase for the DOA Estimation Stage:

1) Evaluate the correlation matrix of the nth array output vector $\{R^n, n = 1, 2, \cdots, N\}$ using equation (9).
2) Form the vectors $\{\boldsymbol{b}^n, n = 1, 2, \cdots, N\}$.
3) Normalize the input vectors using equation (10).
4) Generate input output pairs $\{\boldsymbol{z}^n, \Theta^n, n = 1, 2, \cdots, N\}$.
5) Employ an appropriate RBFNN training procedure to learn the training set generated in step 4).

2) Generalization (Testing) Phase for the DOA Estimation Stage:

1) Evaluate the sample correlation matrix using the collected array output measurements and equation (9).
2) Form the vectors $\hat{\boldsymbol{b}}$.
3) Produce the normalized input vectors $\hat{\boldsymbol{z}}$ from equation (10).
4) Present input vectors $\hat{\boldsymbol{z}}$ to the RBFNN and obtain a vector of values between 0 and 1 whose further processing will give you an estimate $\hat{\Theta}$.

IV. Results

A linear array of ten elements was trained to detect the presence of sources in a 10° wide sector. Different training and testing sets were generated from sources with equal SNR of 10 dB. The correlation matrix was calculated from 400 snapshots of simulated array measurements. In Fig. 3, the array was trained to detect the presence of two sources separated by 2° in space and then was tested with four sources in the sector with the same angular separation. The actual output of the detection stage shows the ability of the network to generalize and detect more sources than it was trained for. To investigate the behavior of the network for different angular separations a ten-element array was trained with two sources of 2° angular separation and tested with three sources of 3° angular separation. Fig. 4 shows a comparison between the desired and actual response for this array with sources with different SNR, respectively.

Choosing sectors 19° wide (θ_W), and minimum angular resolution $(\Delta\Theta_{\min})$ of 2°, the dimension of the output layer of

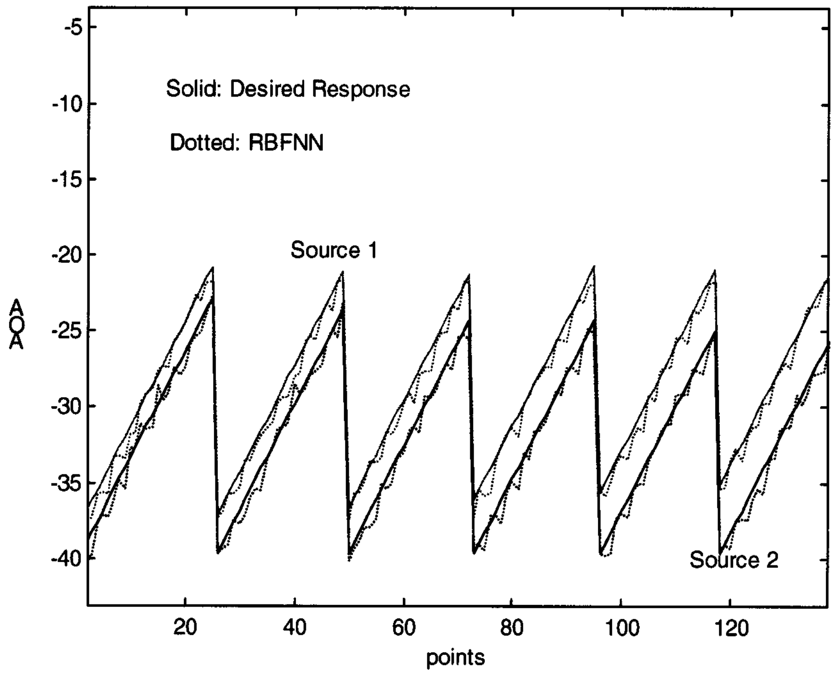


Fig. 7. An array of ten elements is shown tracking two sources in the sector $[-40° \; -21°]$ with angular separations $2° \; -4.5°$ and different sets of random SNR in the training and testing phases.

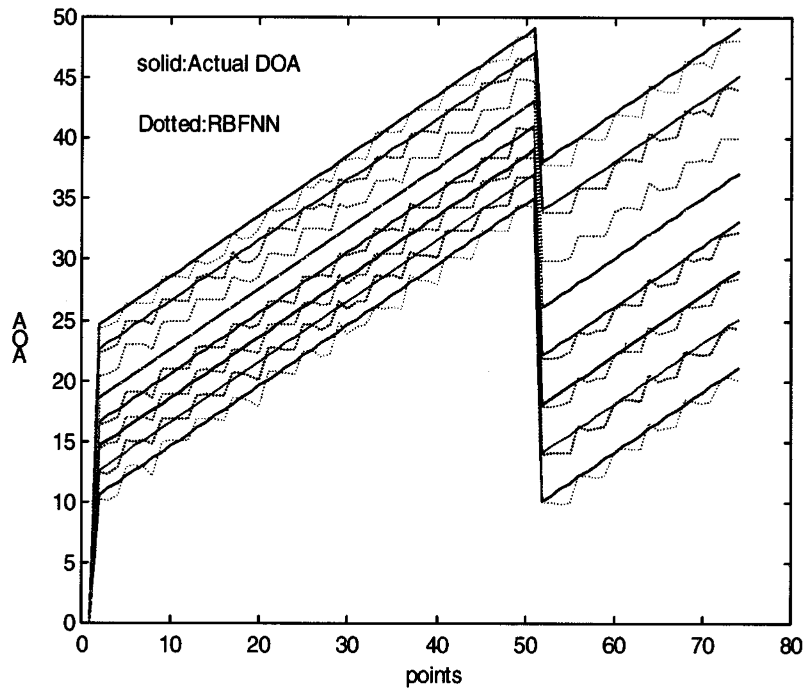


Fig. 8. Comparison between the actual and the DOA estimated by the neural network for a four-element linear array receiving eight sources located in the sector $[10° \; 49°]$ with $2°$ and $4°$ angular separation and random SNR.

individual networks in the estimation stage becomes ten nodes. Fig. 5 shows a linear array of eight elements $(d = \lambda/2)$ tracking four sources of $2°$ angular separation in the sector $[-30° \; -11°]$. The input layer consisted of 72 nodes and the sources were assumed to be of equal power—5 dB higher than the noise power. The estimated and the theoretical angles of arrivals were very close. Since in practice, due to some tuning imperfections or Doppler spread, the operating frequency often changes, a 12-element array was trained with d/λ ranging from 0.4 to 0.6 and with three sources $4°-7°$ of angular separation in the sector $[10° \; 29°]$. The number of points in the abcissa refers to the index of the testing set. This is true for Figs. 6–8 as well. Fig. 6 shows that the RBFNN was able to estimate the DOA of the sources accurately. The dimension of the input layer in this case was 156 nodes. Since the sources do not normally have the same power, in Fig. 7 an array of ten elements is shown tracking two sources in the sector $[-40° \; -21°]$ with angular separations $2° \; -4.5°$ and different sets of random SNR in the training and testing phases. It is observed that the RBFNN successfully resolved those sources. To study the performance of the algorithm when the number of signals is larger than the number of the array elements, Fig. 8 compares the actual and the DOA estimated by the neural network for a four-element linear array receiving eight sources located in the sector $[10° \; 49°]$ with $2°$ and $4°$ angular separation and random SNR. The ability of the network to determine the angles of arrival of a number of sources that is greater than the number of array elements M may be interpreted by the fact that unlike signal subspace based algorithms, (e.g., MUSIC algorithm) no eigendecomposition is necessary and no search is performed in a subspace with dimension less than M.

It should be mentioned here that in all examples presented above, isotropic elements we used. This allows us to assume broad patterns which do not affect the response across the sectors. Also, although we did not experiment extensively with different noise levels, our experience has been that if we train the network with noisy data, it will respond satisfactorily to noisy test data. Finally, in all cases, a network needs an average of 1–15 minutes to train for the DOA estimation stage and about 5 min for the detection stage.

V. Conclusion

A new algorithm is presented for locating and tracking the angles of arrival of multiple sources. This algorithm is based on a family of neural networks operating in two distinct stages. The new approach is based on dividing the field of view of the antenna array into spatial sectors, then each network is trained in the first stage to detect signals emanating from sources in that sector. According to the outputs of the first stage, one or more networks of the second stage can be activated so as to estimate the exact location of the sources. No *a priori* knowledge is required about the number of sources, and the networks can be designed to arbitrary angular resolution. The results demonstrated the high accuracy of the algorithm. The main advantage of this new technique is a dramatical reduction in the size of the training set since much fewer training possibilities need to be considered by sectoring the antenna field of view. It was also demonstrated that neural network based direction finding algorithms possess the ability of locating sources that are greater than the number of the array elements.

References

[1] T. Gebauer and H. G. Gockler, "Channel—Individual adaptive beamforming for mobile satellite communications," *IEEE J. Selected Areas Commun.*, vol. 13, pp. 439–448, Feb. 1995.

[2] S. Swales, M. Beach, D. Edwards, and J. Mcgeehan, "The performance enhancement of multibeam adaptive base-station antennas for cellular land mobile radio systems," *IEEE Trans. Veh. Technol.*, vol. 39, pp. 56–67, Feb. 1990.

[3] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. 34, pp. 276–280, Mar. 1986.

[4] H. L. Southall, J. A. Simmers, and T. H. O'Donnell, "Direction finding in phased arrays with a neural network beamformer," *IEEE Trans. Antennas Propagat.*, vol. 43, p. 1369, Dec. 1995.

[5] L. Long and L. Y. Da, "Real-time computation of the noise subspace for the MUSIC algorithm," in *Proc. ICASSP*, vol. I, 1993, pp. 485–488.

[6] D. Goryn and M. Kaveh, "Neural networks for narrowband and wideband direction finding," in *Proc. ICASSP*, 1988, pp. 2164–2167.

[7] A. H. El Zooghby, C. G. Christodoulou, and M. Georgiopoulos, "Performance of radial basis function networks for direction of arrival estimation with Antenna Arrays," *IEEE Trans. Antennas Propagat.*, vol. 45, pp. 1611–1617, Nov. 1997.

[8] S. Haykin, *Advances in Spectrum Analysis and Array Processing*, S. Haykin, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1995, vol. III.

[9] ——, *Neural Networks—A Comprehensive Foundation*. New York: Macmillan, 1994.

[10] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Processing Mag.*, vol. 13, pp. 50–65, Mar. 1996.

[11] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 326–344, 1965.

[12] T. J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computat.*, vol. 1, pp. 281–294, 1989.

[13] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Reading, MA: Addison Wesley, 1976.

Ahmed H. El Zooghby (S'91) was born in Alexandria, Egypt, in 1969. He received the B.Sc. and M.Sc. degrees, both in electrical engineering, from Alexandria University, Egypt, in 1991 and 1994, respectively, and the Ph.D. degree in electrical engineering from the University of Central Florida, Orlando, FL, in 1999.

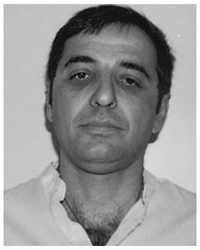
From October 1991 to November 1992 he served as a Communications Systems Engineer in the Air Defense College, Alexandria, Egypt. He joined the Arab Academy for Science and Technology and Maritime Transport in 1992, where he worked as a Lecturer in the Electronics and Computer Engineering Department. He is currently with Qualcomm, Inc., San Diego, CA, working on the Globalstar mobile satellite communication system. His research interests include smart antennas, neural network applications in antennas, adaptive arrays and direction finding for mobile position determination, and mobile satellite communications.

Dr. El Zooghby is a member of Eta Kappa Nu honor Society.

Christos G. Christodoulou (S'80–M'84–SM'90) received the B.Sc. degree in physics and math from the American University of Cairo, Egypt, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from North Carolina State University, Raleigh, in 1981 and 1985, respectively.

He served as a Faculty Member at the University of Central Florida, Orlando, from 1985 to December 1998, where he received numerous teaching, research, and student advising awards. He is currently the Chair of the Electrical and Computer Engineering Department, University of New Mexico, Albuquerque He has over 125 refereed journal publications and conference papers. He also has several book chapters and three patents. He is coeditor of the "Wireless Corner," a column that appears in the *IEEE Antennas and Propagation Magazine*, focusing on both antennas and propagation as they relate to wireless communications. His research interests are in the areas of modeling of electromagnetic systems, neural network applications in electromagnetics, and smart antennas in wireless communications.

Dr. Christodoulou served as the general chair of the IEEE APS/URSI 1999 Symposium, Orlando, FL, and is serving as the cochair of the 2000 IEEE AP-S Conference on Antennas and Propagation for Wireless Communications. In 1991 he was selected as the AP/MTT Engineer of the Year (Orlando Section). He served as Secretary, Treasurer, and Vice President of the Orlando IEEE Section between 1995–1998. He is a member of URSI (Commission B) and the Electromagnetics Academy, SPIE, and ASEE.



Michael Georgiopoulos received the Diploma degree in electrical engineering from the National Technical University, Athens, Greece, in 1981, and the M.S. and Ph.D degrees from the Department of Electrical Engineering, University of Connecticut, Stors, in 1983 and 1986, respectively.

In 1987, he joined the University of Central Florida, Orlando, where he is currently an Associate Professor in the Department of Electrical and Computer Engineering. His research interests are in the areas of neural networks, fuzzy logic, genetic algorithms, and pattern recognition. He is also interested in applications of the aforementioned technologies in communications, electromagnetics, signal/image processing, forecasting, etc.

Dr. Georgiopoulos is a member of the Technical Chamber of Greece and of the International Neural Network Society.