Multiple Source Angle of Arrival Estimation Using Neural-Network Based Smart Antennas

C. G. Christodoulou*, and M. Georgiopoulos, and A.H.EL Zooghby*****,

* Electrical and Computer Engineering Department, University of New Mexico, Albuquerque, NM. 87131
** School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL. 32816
*** Qualcomm, Inc. San Diego, CA

ABSTRACT

The Neural Multiple Source Tracking (N-MUST) algorithm, which is based on an architecture of a family of radial basis function neural networks (RBFNN), is investigated for multiple source tracking with neural network-based adaptive array antennas. The N-MUST algorithm consists of an initial stage with a number of Radial Basis Function Neural Networks (RBFNN) trained to detect the presence of the sources, while a second stage of networks is trained to estimate the exact locations of the sources. The field of view of the antenna array is divided into separate 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 networks are activated to perform the direction of arrival (DOA) estimation step. No prior knowledge of the number of present sources is required. Simulation results are performed and experimental data is applied to the networks to investigate the required training criteria to achieve good generalization in the detection mode, with respect to the angular separations and relative SNR of the sources. The results show substantial reduction in the computational complexity of the network training compared to the single network approach.

Keywords

Smart antennas, source arrival estimation, neural networks, radial basis function neural networks, unsupervised learning, supervised learning.

I. INTRODUCTION

Recently, neural networks-based direction finding algorithms have been proposed for single and multiple source direction finding [1-3]. It has been shown that neural networks have the capability to track sources in real-time. In [4], a radial basis function neural network has been used to track the locations of mobile users. However different networks were needed for different number of users with fixed angular separation. This paper presents a generalization of the algorithm introduced in [4] 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 architecture of a family of radial basis function neural networks that perform both detection and direction of arrival (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 architecture 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. 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 described. 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 i^{th} array element can be written as,

In *Digital Wireless Communication II*, Raghuveer M. Rao, Soheil A. Dianat, Michael D. Zoltowski, Editors, Proceedings of SPIE Vol. 4045 (2000) • 0277-786X/00/$15.00

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

where $s_m(t)$ is the signal of the m^{th} wave, $n_i(t)$ is the noise signal received at the i^{th} 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:

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

Where, **X** (t), **N**(t) and **S**(t) are given by:

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

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

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

In (4) and (5) and (6) the superscript "T" indicates the transpose of the matrix. Also in (3) **A** is the MxK steering matrix of the array towards the direction of the incoming signals defined as:

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

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

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

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 **S**(t) , then the received spatial correlation matrix, **R**, of the received noisy signals can be expressed as:

$$\mathbf{R} = E\{\mathbf{X}(t)\mathbf{X}(t)^H\} = \mathbf{A}\,E[\mathbf{S}(t)\mathbf{S}^H(t)]\mathbf{A}^H + E[\mathbf{N}(t)\mathbf{N}^H(t)] \tag{9}$$

In the above equation, "H" denotes the conjugate transpose. The antenna array can be thought of as performing a mapping G: $\mathbf{R}^K \rightarrow \mathbf{C}^M$ from the space of the DOA, $\{\Theta = [\theta_1, \theta_2, \cdots, \theta_K]^T\}$ to the space of sensor output $\{\mathbf{X}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_M(t)]^T\}$. 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 a Radial Basis Function Neural Network can approximate an arbitrary function from an input space of arbitrary dimensionality to an output space of arbitrary dimensionality[5-7]. 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. The RBFNN is designed to perform an input-output mapping trained with examples. The purpose of the hidden layer in a RBFNN is to transform the input data from an input space of some dimensionality to a space of higher dimensionality *L*. The rationale behind this transformation is based on Cover's theorem[8] which states that an input/output mapping problem cast in a high-dimensionality space nonlinearly is easier to solve. The nonlinear functions 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 [9], where an unsupervised learning algorithm, such as the K-Means[10], 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 problem. For more details about the delta rule, and how it is applied to single layer networks see[6] . 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.

II.1. Sample Data Preprocessing

The input vector to the input layer of the network is the upper triangular half of the spatial correlation matrix **R** that can be organized as an M(M+1) dimensional vector of real and imaginary parts denoted **b**. This procedure is illustrated in Table 1. 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.

$$\mathbf{z} = \frac{\mathbf{b}}{\|\mathbf{b}\|} \quad (10)$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad \mathbf{b} = [r_{11} \quad r_{12} \quad r_{13} \quad r_{22} \quad r_{23} \quad r_{33}]$$

Table 1. Correlation matrix reduction

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. The exhaustive training involved 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. THE NEURAL MULTIPLE SOURCE TRACKING (N-MUST) ALGORITHM

The Neural Multiple Source Tracking (N-MUST) algorithm is also based on the radial basis function neural networks (RBFNN), but it is composed of two stages, the *detection* stage and the *estimation* stage, as shown in Figure 1. In the first stage, a number of RBFNNs 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 sources present is required.

III.1 Detection Stage

In this approach, labeled the Neural Multiple Source Tracking (N-MUST) algorithm, the antenna array can track an arbitrary number of mobile users (sources) without prior knowledge of the number of mobile users. As shown in Figure 1, there are two stages of RBFNN's. The first stage is the "detection stage" which consists of P RBFNNs, each focusing on a sector of width θ_W. The entire angular spectrum (field of view of the antenna array) is divided in P sectors. The p^{th} ($1 \leq p \leq P$) 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 1 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 **R** and one output node. The number of hidden nodes in the second layer is also M (M+1). 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 $[10^0\ 20^0]$ sector with some angular separation $\Delta\theta$. We start the training with sources at -90^0, $-90^0+\Delta\theta,\ldots, -90^0+(N_s-1)\Delta\theta$. We use this vector of DOA to generate the correlation matrix **R** and the normalized vector **z**. We then select the subsequent DOA vectors as -88^0, $-88^0+\Delta\theta,\ldots, -88^0+(N_s-1)\Delta\theta$, -86^0, $-86^0+\Delta\theta,\ldots, -86^0+(N_s-1)\Delta\theta$ 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 $[10^0\ 20^0]$ range; otherwise the target output of the network is zero. 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.

III.2. 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 Figure 1). 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 **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). Consider a system with minimum source resolution of 2^0, a single neural network trained to track sources over the antenna's field of view (e.g.180^0 wide) could be trained for angular separations $\Delta\theta$ of $2^0,4^0,6^0$, ...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 10^0-20^0 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".

IV. SIMULATIONS AND RESULTS

To investigate the behavior of the network for various angular separations a linear array of 10 elements was trained to detect the presence of sources in the 10^0 wide sector [-35^0 -25^0]. Different training and testing sets were generated from 2 sources of 2^0 angular separation and 4 sources of 2^0 angular separation, respectively. The correlation matrix was calculated from 400 snapshots of simulated array measurements. Figure 2 shows a comparison between the desired and actual response for this array with sources having a SNR of 10 dB. For Emergency 911 services, the FCC regulations require a base station to be able to locate a mobile user with an accuracy of 125m (400feet) 67% of the time. To investigate the accuracy of the N-MUST algorithm in localizing users, we simulated a scenario with sectors 19^0 wide (θ_W), and minimum angular resolution ($\Delta\theta_{min}$) of 2^0. In this case, the dimension of the output layer of individual networks in the estimation stage becomes 10 nodes. Figure 3 and 4 show the error in the DOA estimates of 4 sources with 2^0 angular separation in the sector [-30^0 -11^0] using a linear array of 8 elements ($d=\lambda/2$) assuming cells 3 and 5 miles wide, respectively. The input layer of a single neural network consisted of 72 nodes and the sources were assumed to be of equal power, 5 dB higher than the noise power. In Figure 5 and 6, the estimation errors of the same array tracking 2 sources with different angular separations with densely populated cells of 3 and 2 mile diameters, respectively, is shown. In both cases, the N-MUST algorithm achieves a better accuracy than the FCC requirements.

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 2 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.

VI. ACKNOWLEDGEMENT

This work was partly funded by the Florida Space Grant Consortium and Neural Ware Inc.

VII. REFERENCES

1. H.L.Southall, J.A.Simmers, and T.H.O'Donnell," Direction finding in phased arrays with a neural network beamformer", *IEEE Transactions on Antennas and Propagation*, Vol. 43, No. 12, pp. 1369, December 1995.
2. Luo Long, Li Yan Da, " Real-time computation of the noise subspace for the MUSIC algorithm", *Proceedings ICASSP* 1993, Vol I, pp.485 -488, 1993.
3. D.Goryn and M.Kaveh, " Neural networks for narrowband and wideband direction finding", *Proceeding ICASSP*, pp.2164-67, 1988.
4. El Zooghby A. H., C.G. Christodoulou and M. Georgiopoulos," Performance of radial basis function networks for direction of arrival estimation with Antenna Arrays", *IEEE Transactions on Antennas and Propagation*, Vol.45, No.11, pp.1611-1617, November 1997.
5. S.Haykin, editor, *Advances in Spectrum Analysis and Array Processing*, Vol. III, Prentice Hall, NJ, 1995.
6. S.Haykin, *Neural Networks A Comprehensive Foundation*, Macmillan College Publishing, 1994.
7. B. Mulgrew, " Applying Radial Basis Functions", *IEEE Signal Processing magazine*, Vol 13, No.2, pp.50-65, March 1996.
8. T.M.Cover," Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition", *IEEE Transactions on Electronic Computers*, EC-14, pp. 326-344, 1965.
9. T.J.Moody and C.J.Darken, " Fast learning in networks of locally tuned processing units", *Neural Computation*, Vol.1, pp.281-294, 1989.
10. J.T.Tou and R.C.Gonzalez, *Pattern Recognition Principles*, Addison Wesley, Reading, MA, 1976.

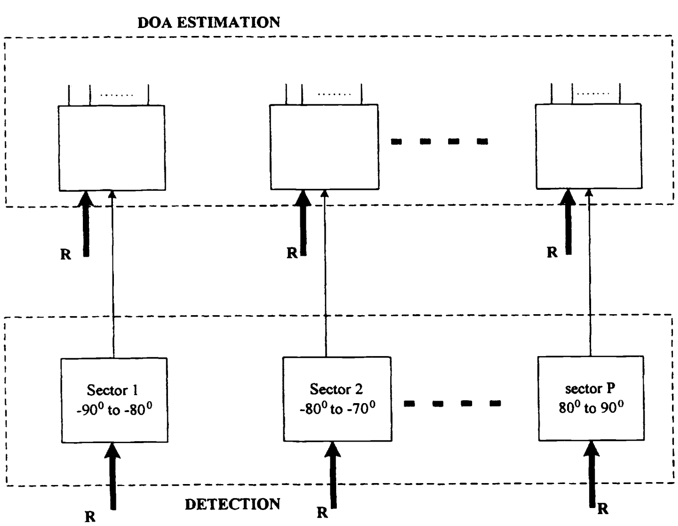

Figure 1 The Neural Multiple Source Tracking architecture.

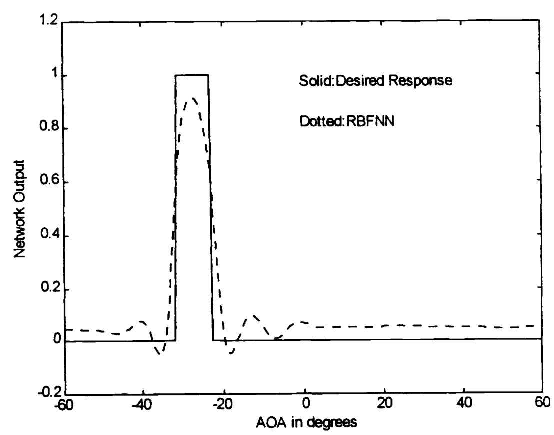

Figure 2 A 10 element linear array and sources in a 10^0 wide sector[-25^0 -35^0]. Trained with 2 sources and tested with 4 sources separated by 2^0 in space.

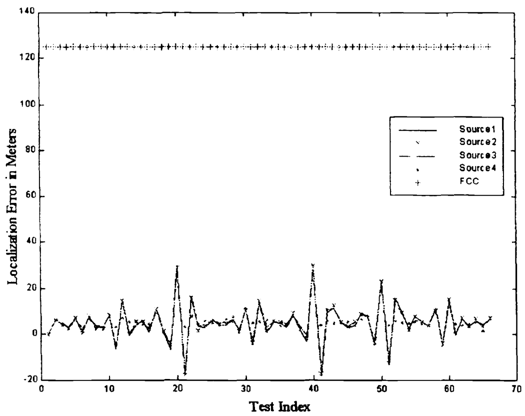

Figure 3 Estimate error ,M=8, 4 sources, 3 mile wide cell

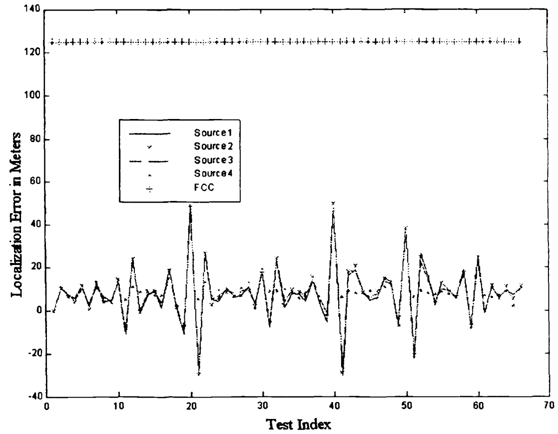

Figure 4 Estimate error ,M=8, 4 sources, 5 mile wide cell

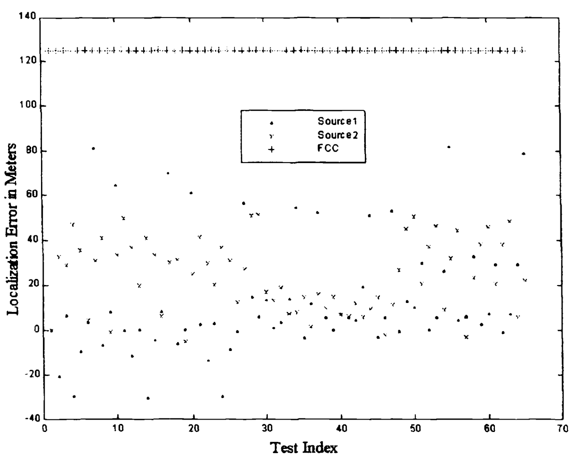

Figure 5 Estimate error ,M=8, 2 sources, 3 mile wide cell

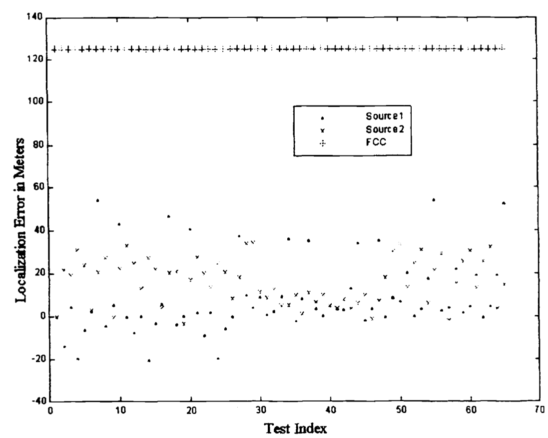

Figure 6 Estimate error ,M=8, 2 sources, 2 mile wide cell