Neural Network Processing for Adaptive Array Antennas

C. G. Christodoulou *
Electrical and Computer Engineering Department
University of New Mexico
Albuquerque, NM 87131

A.H.EL Zooghby and M. Georgiopoulos
Electrical and Computer Engineering Department
University of Central Florida
Orlando, Florida 32816

I. Introduction. Today's wireless systems are required to satisfy an increasing demand for coverage, capacity, and service quality. Advanced signal processing techniques are combined with antenna arrays concepts to produce some promising innovative solutions. Existing wireless systems cannot effectively address problems such as cochannel interference (CCI). Cochannel interference is the most serious limiting capacity factor in any mobile communication system. As the number of users increase, within a certain region, the likelihood of interfering with one another increases. In order to solve the CCI problem, first a superresolution Direction Of Arrival (DOA) algorithm is utilized to locate the desired as well as the cochannel mobile users. Next, an adaptive array antenna can be used to steer its radiation beam towards the mobiles of interest [1] and nulls toward the other sources of interference in the same frequency slot.

Currently, several algorithms can be used to perform the direction finding or angle of arrival of signals from mobile users. One drawback of these algorithms is their difficulty of their implementation in **real-time** because of their intensive computational complexity. Neural networks, on the other hand, due to their high-speed computational capability, can yield results in real-time. Moreover, conventional beamformers require highly calibrated antennas with identical element properties. Performance degradation often occurs due to the fact that these algorithms poorly adapt to element failure or other sources of errors. Neural network-based array antennas do not suffer from this shortcoming.

II. Neural-network based Direction of Arrival Estimation. Both problems, DOA and null steering (beam steering), are approached as a mapping which can be modeled using a suitable artificial neural network trained with input output pairs. The network is then capable of estimating or predicting outputs not included in the learning phase through generalization. Here, the neural network of choice is the radial basis function neural network (RBFNN) [2], shown in Figure 1 with its input preprocessing and output post –processing sections. For the DOA problem, the array performs the mapping G: $\mathbf{R}^K \rightarrow \mathbf{C}^M$ from the space of DOA, $\{\theta = [\theta_1, \theta_2, \cdots, \theta_K]\}$ to the space of sensor output $\{\mathbf{s} = [s_1, s_2, \cdots s_M]\}$ namely:

$$s_m = \sum_{k=1}^{K} a_k e^{j(m\frac{\omega_0}{c} d \sin\theta_k + \alpha_k)} \qquad (1)$$

0-7803-5639-X/99/$10.00 ©1999 IEEE.

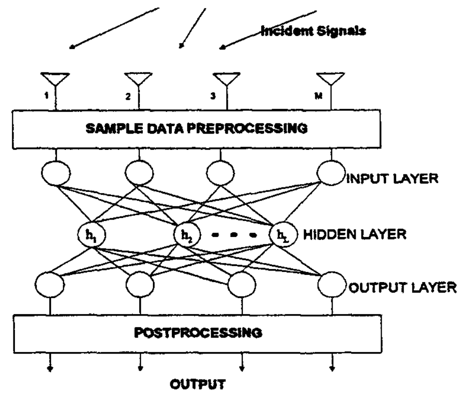

Figure 1. Neural network-based adaptive array processing

That is for each incident signal (Sm) an angle (θm) is associated with it. K is the number of signals, M is the number of elements of a linear array, a_k represents the complex amplitude of the k^{th} signal, α_k the initial phase and ω_0 is the center frequency. A neural network is used to perform the inverse mapping F: $C^M \rightarrow R^K$. The network is trained by N that associate the output vectors s (1), s(2), …, s (N) with the corresponding DOA vectors θ(1), θ(2), …, θ(N). Input vectors s are mapped through the hidden layer of the neural network, then each output node computes a weighted sum of the hidden layer outputs.

The estimation phase consists of transforming the sensor output vector into an input vector and producing the DOA estimate. We can, thus, rewrite the spatial correlation matrix **R** as:

$$R_{mm'} = \sum_{k=1}^{K} p_k e^{\frac{j(m-m')\omega_0 d \sin\theta_k}{c}} + \delta R_{mm'} \qquad (2)$$

where p_k denotes the power of the kth signal. The last term of the right hand side of this equation contains all the cross-correlated terms between signals. Since for m=m', R_{mm} does not carry any information on the DOA ($R_{mm} = \sum_{k=1}^{K} p_k$), we can rearrange the rest of the elements into a new input vector, **b**, which can be defined as:

$$b = \left[R_{21}, \cdots, R_{M2}, R_{12}, \cdots, R_{M2}, R_{1M} \cdots, R_{M(M-1)}\right]^T \qquad (3)$$

The dimension of the hidden layer is equal to the number of the Gaussian functions L that can be chosen to be equal to N if perfect recall is desired. Obviously, the number of output node is equal to the number of signals K. The input vector is then normalized by its norm in the training, testing and estimation phases.

III. Adaptive beamforming. 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 $\{\theta_1 \quad \theta_2 \quad \cdots \quad \theta_K\}$. The elements' output of the array have the form of the M-dimensional vector:

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

and the weights of the element outputs can be represented in the M-dimensional vector:

$$\mathbf{W} = [w_1 \quad w_2 \quad \cdots \quad w_M]^T \tag{5}$$

These weights are also the excitations required to feed the array elements in order to perform the appropriate beam steering. The array output can be written as:

$$y(t) = \sum_{i=1}^{M} w_i^* x_i(t) = \mathbf{W}^H \mathbf{X}(t) \tag{6}$$

It can be shown that the optimum weight vector is given by the following equation [3]:

$$\hat{\mathbf{W}}_{opt} = \mathbf{R}^{-1}\mathbf{S}_d\left[\mathbf{S}_d{}^H\mathbf{R}^{-1}\mathbf{S}_d\right]^{-1}\mathbf{r} \tag{7}$$

Since the above equation is not practical for real time implementation, neural networks can be used to determine the appropriate weights for the desired beam steering and nulling [4].

IV. Results. Figure 2 shows a linear array of 8 elements ($d=\lambda/2$) tracking 4 sources of 2^0 angular separation in the sector $[-30^0 \quad -11^0]$. 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.

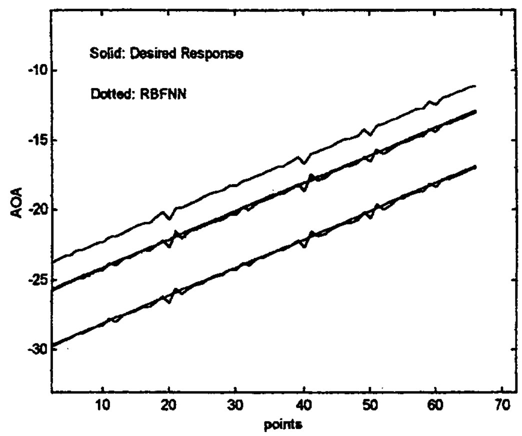

Figure 2. Response of an 8 element linear array ($d=\lambda/2$) tracking 4 sources of 2^0 angular separation

The example in Figure 3 demonstrates the RBFNN implementation of an adaptive array for beamforming and null steering. A 6 element linear array, of $\lambda/2$ spacing between the elements, is used to track 3 signals, two of which are desired and the third is a jammer. The signals are 20^0 apart in space with equal SNR of 20 dB above the noise level. The neural network has 42 input nodes, 42 hidden nodes and 12 output nodes. The adapted pattern obtained from the network is shown (dotted curve) and compared to the optimum pattern obtained from the Wiener solution (solid curve) as the array tracks the mobile signals at different spatial locations.

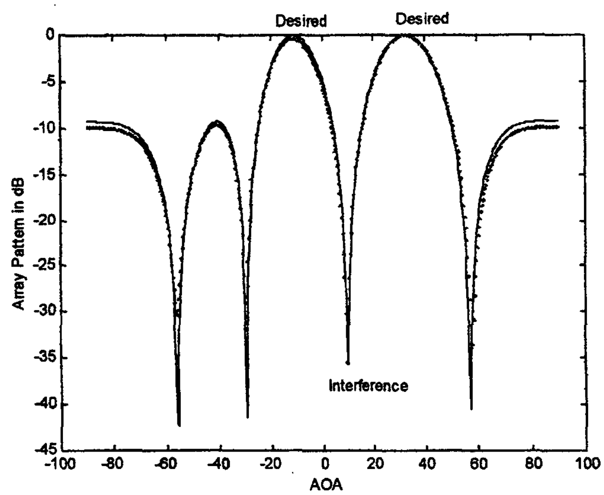

Figure 3 Pattern for an adaptive array antenna tracki ng two desired signals and a jammer

V. Conclusion.. One or more neural networks can be used tracking multiple users while simultaneously nulling interference caused by cochannel users. Both the angles, for the angles of arrival problem, and the weights, for the appropriate null or beam steering, were computed using RBFNNs

V. References

[1] T.Gebauer, and H.G.Gockler, "Channel -individual adaptive beamforming for mobile satellite communications", *IEEE Journal on Selected Areas in Communications*, vol.13, No 2, pp. 439-448 February 1995.

[2] 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 *Trans .on Antennas and Propagation*, vol.45, No.11, pp.1611-1617, November 1997.

[3] Mozingo, Miller, Introduction to Adaptive Arrays, John Wiley, 1980.

[4] El Zooghby A. H., C.G. Christodoulou and M. Georgiopoulos, "Neural Network-Based Adaptive Beamforming for One- and Two-Dimensional Antenna Arrays", IEEE *Trans .on Antennas and Propagation*, vol. 46, No.12, December 1998.