12

ADAPTIVE INTERFERENCE CANCELLATION WITH NEURAL NETWORKS

A.H.EL Zooghby, C. G. Christodoulou, and M. Georgiopoulos
Electrical and Computer Engineering Department
University of Central Florida
4000 Central Florida Blvd, Orlando, Florida 32816
ahe@ece2.engr.ucf.edu

Abstract

In modern cellular, satellite mobile communications systems, and in GPS systems, a fast tracking system is needed to constantly locate the users, and then adapt the radiation pattern of the antenna to direct multiple narrow beams to desired users and nulls to sources of interference. In this paper, the computation of the optimum weight vector of the array is approached as a mapping problem which can be modeled using a suitable artificial neural network trained with input output pairs, A three-layer radial basis function neural network (RBFNN) is used in the design of one and two-dimensional array antennas to perform beamforming and nulling. RBFNN's are used due to their ability to interpolate data in higher dimensions. Simulations results performed under different scenarios of angular separations, and SNR are in excellent agreement with the Wiener solution. It was found that networks implementing these functions are successful in tracking mobile users as they move across the antenna's field of view.

1. Introduction

Multiple access techniques are often used to maximize the number of users a wireless communications system can accommodate. With frequency reuse, where the same frequency is used in two different cells separated far enough so that users in one cell do not interfere with the users in the other cell. Further improvements in the system capacity can be achieved. Moreover,

adaptive arrays implemented in base stations allow for closer proximity of cofrequency cells or beams providing additional frequency reuse by rejecting or minimizing cochannel and adjacent channel interference[l]. Motivated by the inherent advantages of neural networks([2]-[7]), this paper presents the development of a radial basis function neural network-based algorithm to compute the weights of an adaptive array antenna [8]. In this new approach, the adaptive array can detect and locate mobile users, track these mobiles as they move within or between cells, and allocate narrow beams in the directions of the desired users while simultaneously nulling unwanted sources of interference. This paper is organized as follows: In sections 2 an 3 a brief derivation of the optimum array weights in 1-D and 2-D adaptive beamforming is presented. In Section 4 the RBFNN approach for the computation of the adaptive array weights is introduced. Finally, Section 4 presents the simulation results and Section 6 offers some conclusive remarks.

2. Adaptive beamforming using 1-Dimensional linear arrays

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} element can be written as,

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

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 on the matrix forms:

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

Where, X (t), S(t) and N(t) are M-dimensional vectors. Also in (3) A is the MxK steering matrix of the array towards the direction of the incoming signals defined as:

$$\mathbf{A} = \left[\mathbf{a}(\theta_1) \quad \mathbf{a}(\theta_2) \quad \cdots \quad \mathbf{a}(\theta_K)\right] , \tag{4}$$

where $\mathbf{a}(\theta_i)$ is defined as

$$\mathbf{a}(\theta_i) = \left[1 \quad e^{-jk_i} \quad e^{-j2k_i} \quad \cdots \quad e^{-j(M-1)k_i}\right] \tag{5}$$

Classical Approach

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, $\mathbf{R}$, of the received noisy signals can be expressed as:

$$\begin{aligned} \mathbf{R} &= E\{\mathbf{X}(t)\mathbf{X}(t)^H\} = A\,E[\mathbf{S}(t)\mathbf{S}^H(t)]A^H + E[\mathbf{N}(t)\mathbf{N}^H(t)] \\ &= APA^H + \sigma^2 I = \sum_{i=1}^{M} \lambda_i \; e_i e_i^H \end{aligned} \tag{6}$$

In the above equation, $\mathbf{P} = E\{\mathbf{S}(t)\mathbf{S}(t)^H\}$ designates the signal covariance matrix and I is the identity matrix. Also, in the above equation "H" denotes the conjugate transpose. Finally, λ_i $(1 \le i \le M)$ and e_i $(1 \le i \le M)$ stand for the eigenvalues and eigenvectors of the matrix $\mathbf{R}$, respectively. With the weights of the array element outputs represented as an M-dimensional vector $\mathbf{W}$ the array output becomes

$$y(t) = \sum_{i=1}^{M} w_i^* x_i(t) = \mathbf{W}^H \mathbf{X}(t) \tag{7}$$

The mean output power is thus given by:

$$P(\mathbf{W}) = E[y(t)y^*(t)] = \mathbf{W}^H \mathbf{R} \mathbf{W} \tag{8}$$

where * denotes the conjugate. To derive the optimal weight vector, the array output is minimized so that the desired signals are received with specific gain, while the contributions due to noise and interference are minimized. In other words:

$$\min \mathbf{W}^H \mathbf{R} \mathbf{W} \qquad \textit{subject to } \mathbf{W}^H \mathbf{S}_d = \mathbf{r} \tag{9}$$

In the above equation, $\mathbf{r}$ is the V x 1 constraint vector, where V is the number of desired signals, and S_d is the steering vector associated with the look direction as defined in (5). The method of Lagrange multipliers is used to solve the constrained minimization problem in (9). It can be shown that the optimum weight vector is given by the following equation:

$$\hat{\mathbf{W}}_{opt} = \mathbf{R}^{-1}\mathbf{S}_d \left[\mathbf{S}_d{}^H \mathbf{R}^{-1} \mathbf{S}_d\right]^{-1} \mathbf{r} \tag{10}$$

Since the above equation is not practical for real time implementation, an adaptive algorithm must be used to adapt the weights of the array in order to track the desired signal and to place nulls in the direction of the interfering signals. In this work, neural networks are utilized to achieve this adaptive response in real-time, the details of which appear in section 4.

3. Adaptive beamforming using 2-D rectangular arrays

Consider a general M x N rectangular array receiving K signals and let the received signal data matrix be given by [9]:

$$x_{mn}(t) = \sum_{i=1}^{K} s_i(t) A_{ui}(m) A_{vi}(n) + n_{mn}(t) \tag{11}$$

where

$$A_{ui}(m) = \exp\left\{ j(m-1)\frac{2\pi d_u}{\lambda} \sin\theta_i \cos\phi_i \right\} \tag{12}$$

$$A_{vi}(n) = \exp\left\{ j(n-1)\frac{2\pi d_v}{\lambda} \sin\theta_i \sin\phi_i \right\} \tag{13}$$

In the above equation , d_u and d_v are the spacings between the elements along the column and row directions, respectively, while θ_i and ϕ_i are the elevation and azimuth angles of the i^{th} source, respectively, and m=l,2,...,M; n=l,2, ...,N; and i=l,2, ...,K.The received signal data can be arranged in a 1x MN vector given by

$$\mathbf{X}(t) = \sum_{i=1}^{K} s_i(t)\mathbf{A}_i + \mathbf{N}(t) \tag{14}$$

The i^{th} signal direction vector $\mathbf{A}_i = \mathbf{A}_{vi} \otimes \mathbf{A}_{ui}$ is defined in terms of the Kronecker product of $\mathbf{A}_{vi}$ and $\mathbf{A}_{ui}$,which are given by

$$\mathbf{A}_{ui} = \left[A_{ui}(1) \quad A_{ui}(2) \quad \cdots \quad A_{ui}(M)\right]^T \quad \text{and}$$

$$\mathbf{A}_{vi} = \left[A_{vi}(1) \quad A_{vi}(2) \quad \cdots \quad A_{vi}(M)\right]^T \tag{15}$$

It can be shown that in this case, the vector of optimum weights is given by

$$\hat{\mathbf{W}}_{opt} = \mathbf{R}^{-1}\mathbf{S}_d\left[\mathbf{S}_d{}^{\mathrm{H}}\mathbf{R}^{-1}\mathbf{S}_d\right]^{-1}\mathbf{r} \tag{16}$$

where $\mathbf{R}$ is defined as $E\{\mathbf{X}(t)\,\mathbf{X}^{\mathrm{H}}(t)\}$ and $\mathbf{S}_d$ is the steering vector associated with the desired signals given by $\mathbf{S}_d = \mathbf{S}_{dv} \otimes \mathbf{S}_{du}$ where $\mathbf{S}_{dv}$ and $\mathbf{S}_{du}$ are vectors defined as

$$\mathbf{S}_{du} = \left[S_{du}(1) \quad S_{du}(2) \quad \cdots \quad S_{du}(M)\right]^T \tag{17}$$

$$\mathbf{S}_{dv} = \left[S_{dv}(1) \quad S_{dv}(2) \quad \cdots \quad S_{dv}(N)\right]^T \tag{18}$$

whose elements are given as in (12) and (13).

4. Neural Network -based interference cancellation:

This section describes a new implementation for the problem of beamforming using neural networks. Since, optimum weight vector is a nonlinear function of the correlation matrix and the constraint matrix (see equations (10) and (16)). Then it can be approximated using a suitable neural net architecture such as the Radial Basis Function Neural Network [11]. 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 [10]. The block diagram of the RBFNN based array is shown in Figure 1. As it can be seen from Figure 2, the RBFNN consists of three layers of nodes, the input layer, the output layer and the hidden layer. In our application the input to the network is the correlation matrix **R** while the output layer consists of *2M* nodes (1-D case) or *2MN* nodes (2-D case) to accommodate the output vector (i.e., $\mathbf{W}_{opt}$).The RBFNN is designed to perform an input-output mapping trained with examples. There are a lot of learning strategies that have appeared in the literature to train an RBFNN. The one used in this paper was introduced in ([10],[11]), where an unsupervised learning algorithm (such as the K-Means [12]) 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. 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 phase*, the neural network is supposed 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.

Input Preprocessing

First, the array output vectors are generated then transformed into appropriate input vectors to be presented to the network. The estimation phase consists of transforming the sensor output vector into an input vector and producing the DOA estimate. The correlation matrix **R** can be rearranged into a new input vector , **b** , given as

$$b = [R_{11}, \cdots, R_{M2}, R_{12}, \cdots, R_{M2}, R_{1M} \cdots, R_{MM}]^T \quad (19)$$

It was found that by only taking the upper triangular part of **R** as the input, the number of input units is significantly reduced without affecting the interpolation capability of the network. Note that we still need twice as many input nodes for the neural network to accommodate the complex inputs.

5. Simulations

The pattern of an array of 8 elements receiving 1 desired signal and 3 interfering is shown in Figure 2.The SNR of the sources is 10 dB.The input to the network consisted of all the elements of the correlation matrix **R.** Hence the dimension of the input layer is 128 nodes. In Figure 3,an array of 10 elements is simulated under the same conditions with 110 Input nodes where only the upper triangular part of **R** was used as the input. The results show that it is possible to reduce the dimension of the input layer significantly without affecting the interpolation capabilities of the network. In Figure 4,an array of 20 elements is shown tracking 7 signals 4 of which are interference. The SNR of the desired signals were set to 10 dB while those of the interfering signals were set to 20 dB and $\Delta\theta = 5^0$. Finally, Figure 5 shows a 4 x 4 planar array is trained to track 7 signals 3 of which are desired with $\Delta\theta = 12^0$ and $\phi=60^0$.

6. Conclusion

A new approach to the problem of adaptive beamforming was developed. The weights were computed using an RBFNN that approximates the Wiener solution. The network was successful in tracking multiple desired users while simultaneously nulling interference caused by cochannel

users or other sources of interference. Both linear and planar arrays were simulated and the results have been very good in every case. Comparison of the adapted pattern obtained by the RBFNN and the optimum solution proved the high degree of accuracy of this approach.

7. Acknowledgement

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

8. 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] 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.

[3] 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.

[4] A.F.Naguib, A.Paulraj, T.Kailath,"Capacity improvement with base-station antenna arrays in cellular CDMA", *IEEE Transactions Vehc.Technology,* vol. 43, No. 3,pp. 691, August 1994.

[5] Luo Long, Li Van Da, "Real-time computation of the noise subspace for the MUSIC algorithm", *Proc. ICASSP* 1993, vol.1, pp.485 -488, April1993.

[6] D.Goryn and M.Kaveh."Neural networks for narrowband and wideband direction finding", *Proc. ICASSP,* pp.2164-67, April 1988.

[7] P.R.Chang, W.H.Yang, and K.K.Chan,"A neural network approach to MVDR beamforming problem", *IEEE Transactions on Antennas and Propagation,* vol.40, No.3, pp. 313-322, March 1992.

[8] Mozingo, Miller, Introduction to Adaptive arrays, John Wiley, 1980.

[9] S.J.Yu, and J.H.Lee,"Design of Two-Dimensional Rectangular Array Beamformers with partial adaptivity", *IEEE Transactions on Antennas and Propagation*, vol.45, No.l, pp. 157-167, January 1997.

[10] S.Haykin, Neural Networks A Comprehensive Foundation, Macmillan College Publishing, Ontario, 1994.

[11] T.J.Moody and CJ.Darken, "Fast learning in networks of locally tuned processing units", *Neural Computation*, vol.1, 281(1989).

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

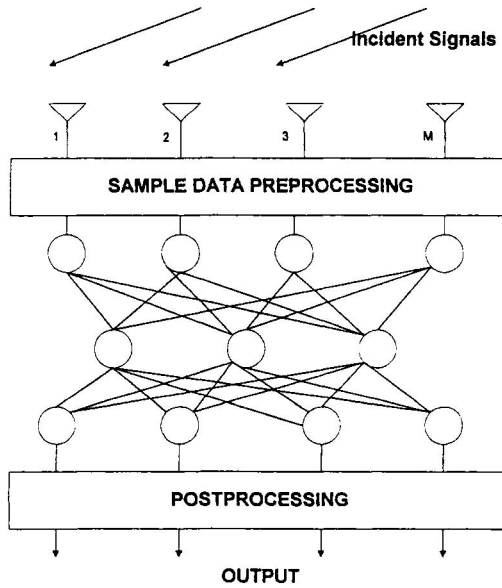

Figure 1 Radial Basis Function Neural Network Architecture

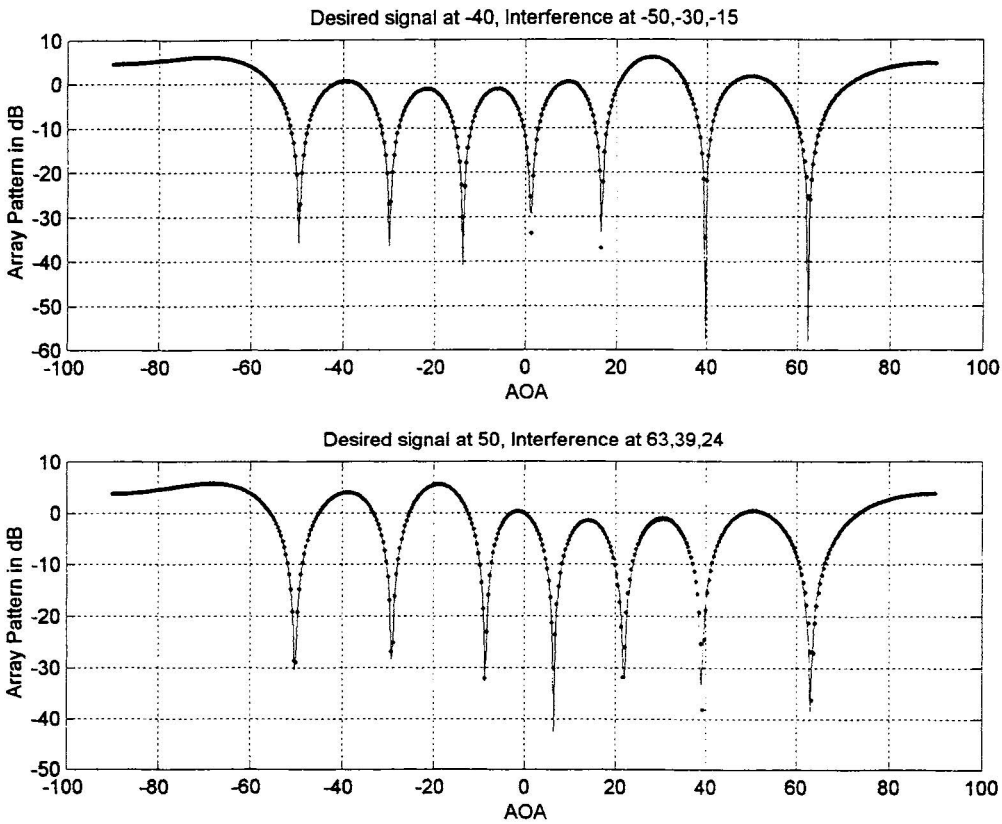

Figure 2 The Pattern of an 8 element array receiving 1 desired signal and 3 interfering signals, SNR is 10 dB, Solid: Wiener, Dotted: rbfnn

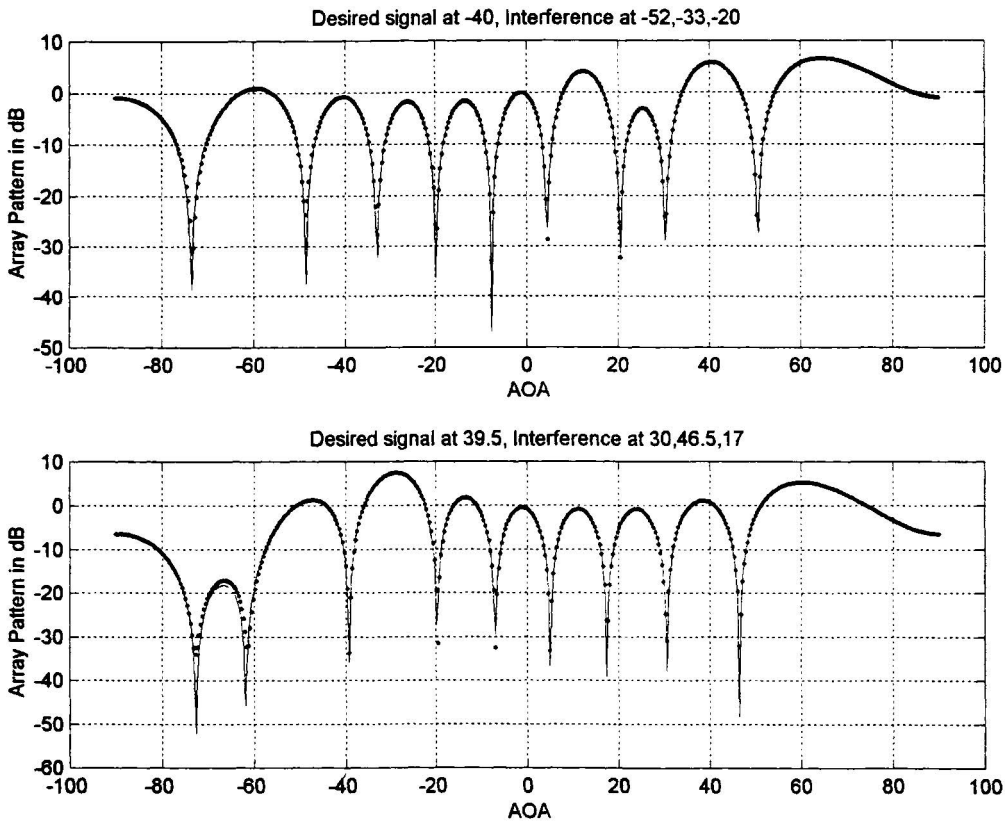

Figure 3 The Pattern of a 10 element array receiving 1 desired signal and 3 interfering signals, SNR is 10 dB, Solid: Wiener, Dotted: rbfnn

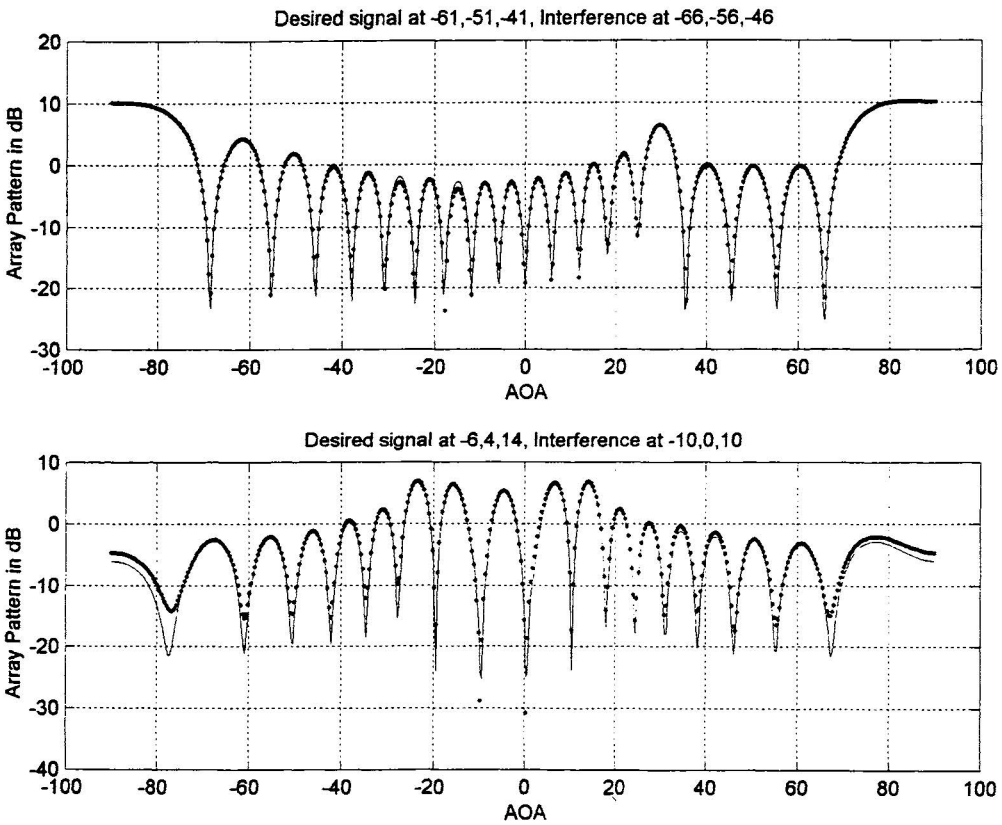

Figure 4 An array of 20 elements tracking 7 signals 4 of which are interference. SNR of desired signals were (10 dB),SNR of interference(20 dB) , and $\Delta\theta = 5^0$. Solid: Wiener, Dotted: rbfnn

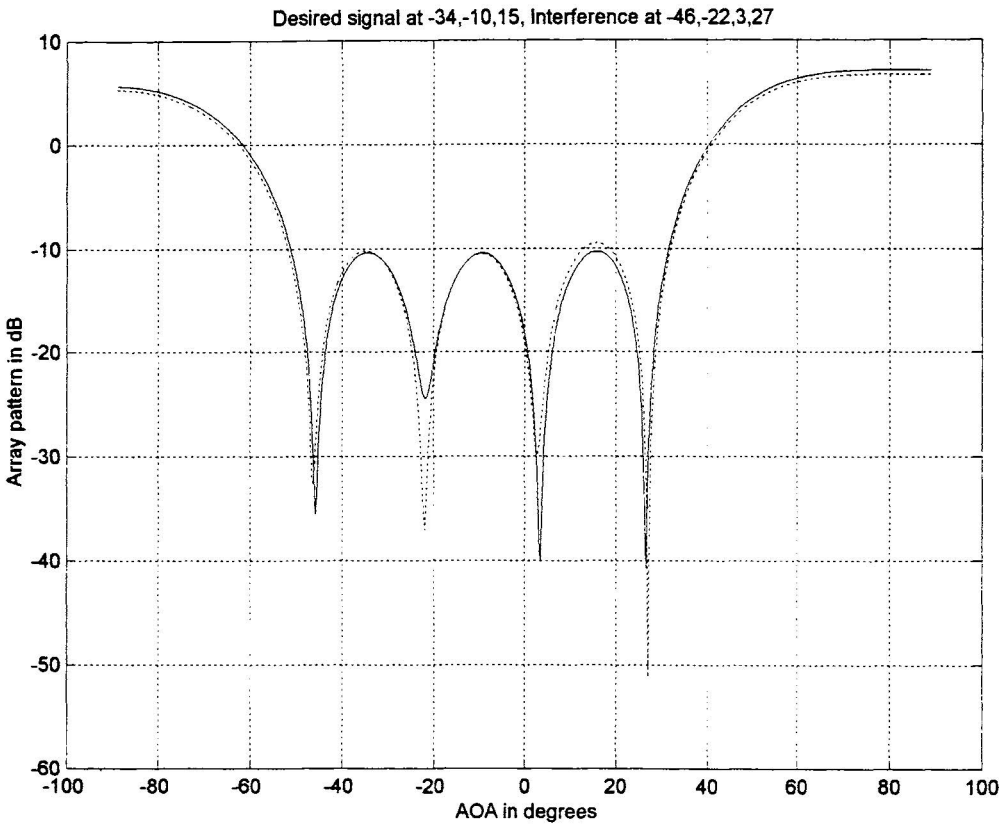

Figure 5 4 x 4 planar array tracking 7 signals 3 of which are desired with $\Delta\theta = 12^0$ and $\phi=60^0$. SNR of desired signals were (10 dB),SNR of interference(20 dB), Solid: Wiener, Dotted: rbfnn