

# Antenna Array Beamforming using Neural Network

Maja Sarevska, and Abdel-Badeeh M. Salem

**Abstract**—This paper considers the problem of *Null-Steering beamforming* using *Neural Network* (NN) approach for antenna array system. Two cases are presented. First, unlike the other authors, the estimated *Direction Of Arrivals* (DOAs) are used for antenna array weights NN-based determination and the imprecise DOAs estimations are taken into account. Second, the blind null-steering beamforming is presented. In this case the antenna array outputs are presented at the input of the NN without DOAs estimation. The results of computer simulations will show much better relative mean error performances of the first NN approach compared to the NN-based blind beamforming.

**Keywords**—Beamforming, DOAs, neural network.

## I. INTRODUCTION

THE field of *artificial* NNs [1,2,3] has made tremendous progress in the past 20 years in terms of theory, algorithms, and applications. Notably, the majority of real world NN applications involve the solution of difficult statistical signal processing problems. Compared to conventional signal processing algorithms [4] that are mainly based on linear model, artificial NNs offer an attractive alternative by providing nonlinear parametric models with universal approximation power, and adaptive training algorithms. In particular, the nonlinear nature of NNs, their ability to learn from their environment in supervised and unsupervised manner, as well as their universal approximation property, makes them highly appropriate for solving difficult signal processing problems.

The research in antenna arrays is most active due to its military and commercial applications, and also in new solutions for future radiotelescopes. The focus of antenna array signal processing is on DOA estimation and beamforming. 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. On the other hand, NN-based antenna array do not suffer from these shortcoming. They use simple addition, multiplication, division, and threshold operations in the basic processing element. They possess advantages as

massive parallelism, nonlinear property, adaptive learning capability, generalization capability, strong fault-tolerant capability and insensitivity to uncertainty.

The paper is organized as follows: Section II describes the DOA estimation problem, Section III presents the NN Null-Steering beamforming based on DOAs, Section IV presents the NN DOA uncertainty problem, the blind Null-Steering beamforming is presented in Section V, Section VI is presenting the results gained from computer simulations, and in Section VII some conclusion remarks are noted.

## II. NN DOA ESTIMATION

Let observe a linear antenna array with M elements, let K ( $K < M$ ) be the number of narrowband plane waves, centered at frequency  $\omega_k$  impinging on the array from directions  $\{\theta_1, \theta_2, \dots, \theta_K\}$ . Using complex signal representation, the received signal in the  $i$ th array element is:

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

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

$$K_m = \frac{\omega_0 d}{c} \sin(\theta_m) \quad (2)$$

where  $d$  is the spacing between the elements of the array, and  $c$  is the speed of the light in free-space. In vector notation the output of the array is:

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

where  $\mathbf{X}(t)$ ,  $\mathbf{N}(t)$ , and  $\mathbf{S}(t)$  are:

$$\begin{aligned} \mathbf{X}(t) &= [x_1(t) \ x_2(t) \ \dots \ x_M(t)]^T \\ \mathbf{N}(t) &= [n_1(t) \ n_2(t) \ \dots \ n_M(t)]^T \\ \mathbf{S}(t) &= [s_1(t) \ s_2(t) \ \dots \ s_K(t)]^T \end{aligned} \quad (4)$$

In (3)  $\mathbf{A}$  is the  $M \times K$  steering matrix of the array toward the direction of the incoming signals:

$$\mathbf{A} = [\mathbf{a}(\theta_1) \ \mathbf{a}(\theta_2) \ \dots \ \mathbf{a}(\theta_K)] \quad (5)$$

where  $\mathbf{a}(\theta_m)$  is the steering vector associated with direction  $\theta_m$ :

$$\mathbf{a}(\theta_m) = [1 \ e^{-jK_m} \ e^{-j2K_m} \ \dots \ e^{-j(M-1)K_m}]^T \quad (6)$$

The received spatial correlation matrix  $\mathbf{R}$  of the received noisy signals can be estimated as:

$$\mathbf{R} = E\{\mathbf{X}(t)\mathbf{X}(t)^H\} = \mathbf{A}E\{\mathbf{S}(t)\mathbf{S}(t)^H\}\mathbf{A}^H + E\{\mathbf{N}(t)\mathbf{N}(t)^H\} \quad (7)$$

Following the Fig.1, the antenna array is performing the mapping  $G: \mathbf{R}^K \rightarrow \mathbf{C}^M$  from the space of DOAs,  $\{\Theta = [\theta_1, \theta_2, \dots, \theta_K]^T\}$  to the space of sensor output  $\{\mathbf{X}(t) = [x_1(t) \ x_2(t) \ \dots$

Maja Sarevska is with the University of Niš, Faculty of Electronic Engineering, Serbia (e-mail: majasarevska@yahoo.com).

Abdel-Badeeh M. Salem is with the Ain Shams University, Faculty of Computer & Information Sciences, Cairo, Egypt (e-mail: absalem@asunet.shams.edu.eg).

$x_M(t)^T$ . A neural network is used to perform the inverse mapping  $F: \mathbf{C}^M \rightarrow \mathbf{R}^K$ . For this task a *Radial Basis Function* NN (RBFNN) is used [5], instead of backpropagation neural network because the second is slower in training. In [5] the values of  $\mathbf{R}$  are used at the input of the NN estimated in block  $\mathbf{R}$  in Fig.1. The antenna view is divided in  $N_N$  sectors and NN learning algorithms for detection and estimation stage are same, the difference is only in the number of nodes in the output layer. Namely, the number of the nodes in the output layer of the first stage (detection) is one (there is a signal gives one, and no signal gives 0), and the number of the nodes in the output layer of the second stage is determined by the angular resolution of the algorithm and the width of the corresponding sector.

There are a lot of learning strategies that have appeared in the literature to train RBFNN. The one used in [5] was introduced in [6], where an unsupervised learning algorithm (such as *K-means* [2]) is initially used to identify the centers of the Gaussian functions used in the hidden layer. The standard deviation of the Gaussian function of a certain mean is the average distance to the first few nearest neighbors of the means of the other Gaussian functions. This procedure allows us to identify the weights (means and standard deviations of the Gaussian functions) from the input to the hidden layer. The weights from the hidden layer to the output layer are estimated by supervised learning known as delta rule, applied on single layer networks [3]. With this procedure, for training we need 5min in detection stage and about 15min in estimation stage. An alternative is instead of using the same neural networks in both stages, to use different neural network in the first stage. The reason for this is the fact that the task of signal detection is a vector classification problem. Any input vector should be classified as 0 (there is NO signal in the corresponding sector) or 1 (there IS a signal in the corresponding sector). For this task an appropriate neural network is *Probabilistic Neural Network* (PNN), which is proposed in [7].

### III. NN-BASED NULL-STEERING BEAMFORMING

Let  $\mathbf{a}(\theta_i)$  be the steering vector in the direction where unity response is desired, and that  $\mathbf{a}(\theta_2), \mathbf{a}(\theta_3), \dots, \mathbf{a}(\theta_K)$  are  $K-1$  steering vectors of interference signal directions. We are trying to put nulls in these  $K-1$  directions and to receive the signal from direction  $\theta_j$ . We can create the antenna radiation pattern by associating a weight value to each antenna element. The desired weight vector is the solution to the following equations:

$$\mathbf{w}^H \mathbf{a}(\theta_j) = 1 \quad (8)$$

$$\mathbf{w}^H \mathbf{a}(\theta_i) = 0, \quad i=2, \dots, K \quad (9)$$

Using matrix notation this becomes:

$$\mathbf{w}^H \mathbf{A} = \mathbf{e}^T \quad (10)$$

where  $\mathbf{e}$  is a vector with all zeros except the first element which is one:

$$\mathbf{e} = [1 \ 0 \ \dots \ 0]^T \quad (11)$$

For  $K=M$ ,  $\mathbf{A}$  is square matrix. Assuming that the inverse of  $\mathbf{A}$  exists, which requires that all steering vectors are linearly independent, the solution for weight vector is:

$$\mathbf{w}^H = \mathbf{e}^T \mathbf{A}^{-1} \quad (12)$$

When steering vectors are not linearly independent  $\mathbf{A}$  is not invertible and its pseudo inverse can be used. Observing the Eq.(12) it follows that the first row of the inverse of  $\mathbf{A}$  forms the desired weight vector.

When the number of required nulls is less than  $M$ ,  $\mathbf{A}$  is not square matrix. A suitable estimate of weights may be produced using:

$$\mathbf{w}^H = \mathbf{e}^T \mathbf{A}^H (\mathbf{A} \mathbf{A}^H)^{-1} \quad (13)$$

RBFNN can successfully perform this Beamforming (BF) procedure and it is presented with block BF RBFNN in Fig.1. Unlike the other authors who use  $\mathbf{R}$  at the input of the NN, in our case we use the DOAs at the input of the BF RBFNN. Given combination of DOAs correspond to given radiation pattern (antenna weight vector) that produce unity response in desired direction, since the NN is trained to give unity response only for one DOA (let say the first one). For multi-user detection we can divide the time into  $K$  slots, and each slot will correspond to one user. In  $k$ -th time slot the position of the desired signal direction:  $\theta_k$ , in the input vector is first one. This time division multiplexing is synchronized with the antenna array.

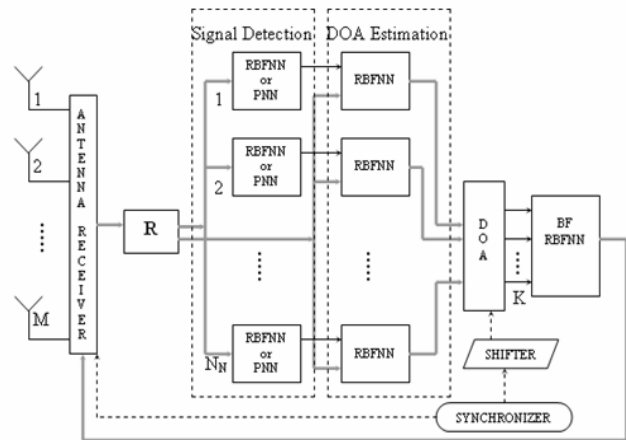


Fig. 1 The Block Diagram of NN-Based Smart Antenna

The BF RBFNN same as in the DOA estimation stage, is consisted of three layers of neurons. The input layer has  $K$  neurons and the number of neurons in the output layer is  $2M$  corresponding to real and imaginary parts of the weights of the antenna array elements. The hidden layer dimension is larger than that of the input layer. The NN weights from the input to the hidden layer are determined by mentioned *K-means* algorithm and the NN weights associated to the neuron connections from the hidden to the output layer are determined with NN training using *delta learning rule*. The BF RBFNN receives input vectors as combinations of DOAs and produces the antenna element weights at the output. Training pairs are produced using Eq. (12). In this case, dividing the space into sectors cannot perform the reduction of the number of training

samples. The reason for this is the fact that the antenna element weights are associated to the whole antenna view. Some other means must be developed in order to decrease the number of training samples. Also as discussed in [8,9] limitation should be expected and future interest is to solve these limitations in order a large number of users to be served.

#### IV. NN DOA UNCERTAINTY PROBLEM

DOA estimation using NN concept is related to some degree of uncertainty. Namely, the actual vector of DOAs:  $\Theta=[\theta_1, \theta_2, \dots, \theta_k]^T$  is presented with estimated vector:  $\Theta'=[\theta_1', \theta_2', \dots, \theta_k']^T$  where:

$$\theta_i' = \theta_i + \Delta\theta_i, \quad i=1,2,\dots,K \quad (14)$$

The parameter  $\Delta\theta_i$  receives random values with uniform distribution in the interval  $[-\text{maxerr}, \text{maxerr}]$ , where  $\text{maxerr}$  is maximal angle error in degrees. This maximal error is dependant from the performances of the NN concept in the DOA estimation phase. It is very important to found out the degree of accuracy that is necessary for DOA estimation in order satisfactory beamforming to be performed. This DOA uncertainty can be decreased by appropriate NN training in DOA estimation phase or by additional training in NN beamforming stage, which will probably overburden the total training in beamforming stage. In the sixth section the worst case will be analyzed, that is when all DOAs are assumed to be imprecise.

#### V. BLIND NULL-STEERING BEAMFORMING

According to worldwide literature an interesting approach of smart antenna concept realization is blind beamforming. The scientists have paid much attention to this approach mainly because of the fast antenna system response. That is why this section will briefly explain the idea of blind Null-Steering beamforming using NN. Namely, the aim in this approach is to perform beamforming without knowing the locations of users. As explained in previous sections, one way to perform beamforming is to estimate DOAs of users and than according to Null-Steering algorithm to estimate the antenna element weights[10]. This approach is realized in two steps: DOAs estimation and beamforming.

Blind beamforming on the other hand do not need DOAs estimations and directly according to the antenna array outputs estimates the weights that should be associated to each antenna element in order a desired radiation pattern to be achieved.

First, a training set of N pairs: [sample, target] must be generated. The samples are generated according to (7) and targets are estimated according to (12). RBFNN is used and preprocessing block that will provide the connection between the antenna array and NN and will estimate the values of the received spatial correlation matrix  $\mathbf{R}$ .

As known the NN is consisted of three layers of neurons with Gaussian transfer functions in the hidden layer. The means and variances of Gaussian functions are estimated using the *K-means* algorithm and the NN weights from the

hidden to the output layer are estimated using the *delta rule* algorithm.

The time needed for training the NN beamformer using the DOAs is much larger since we need to train NNs in three stages: signal detection, DOA estimation, and beamforming (Fig.1). Also we should take into account imprecise DOAs estimations. On the other hand observing the high nonlinear relation between the  $\mathbf{R}$  and antenna element weights for blind beamforming it is a questionable is the NN able to perform the input-output mapping successfully. It is obvious that for blind beamforming we need much more training samples.

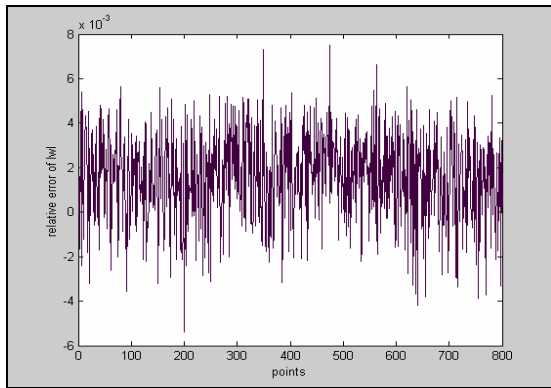
After the training the response of the two systems will be fast since the NN has a very small response time. Computer simulations in the next section will show much better relative error performances of the DOAs based beamforming over the blind, since the RBFNN will have a difficulty to "understand" the nature of the mapping  $\mathbf{R} \rightarrow$  weights. Three factors have influence on  $\mathbf{R}$  values: locations of users, noise, and the combination of transmitted bits. The performances might be improved by modification of the RBFNN or of the learning algorithm. Another possibility is to solve the limitations of RBFNN [8] and to provide the possibility much more training samples to be used. However this issues should be further investigated.

#### VI. COMPUTER SIMULATIONS

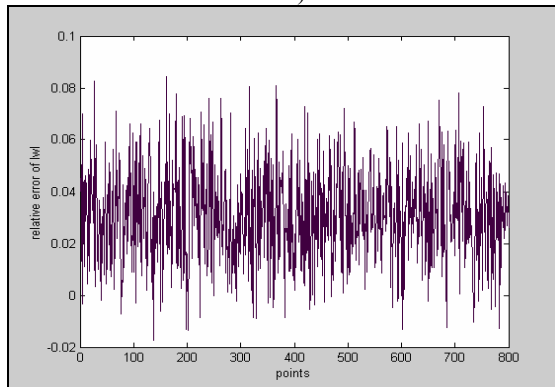
Many different examples were investigated, here the results for the example when there are  $K=6$  users and  $M=6$  antenna elements are exposed. A regular linear antenna array was used with inter-element spacing of  $d=0.5$  wavelengths. The BF RBFNN has 6 neurons in the input layer, 30 in the hidden and 12 neurons in the output one. The centers of Gaussian transfer functions in the hidden layer were determined with *K-means* clustering algorithm. The variances were estimated as the mean distance of the three nearest neighboring centers from the corresponding center. The case for  $\phi=\text{const.}$  and  $\theta \in (0^\circ \div 180^\circ)$  was analyzed. The users were placed in the space with mutual distance of  $20^\circ$ .

Let assume that there are six users at mutual distance of 20 degrees and that the DOA estimation NN has performed the DOA estimation with accuracy within the range of  $[-\text{maxerr}, \text{maxerr}]$ . Fig. 2 is presenting the relative error of the absolute value of the estimated array weights and Fig.3 is presenting the relative error of the argument of the estimated array weights. It can be concluded that null-steering NN successfully performs the beamforming almost completely neglecting the DOA uncertainty for the case when  $\text{maxerr}=0.1^\circ$ . For the case when  $\text{maxerr}=0.5^\circ$ , the influence of DOA uncertainty is obvious. It can be easily concluded that further enlargement of the DOA imprecision (the higher value of  $\text{maxerr}$ ) will largely damage the null-steering NN beamforming performances.

We should mention that we have analyzed the case when the relative error due to imperfect NN beamforming generalization is almost zero in order to observe only the influence of the DOA estimation uncertainty.

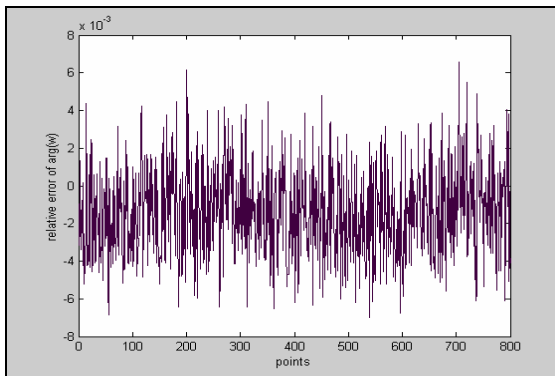


a)

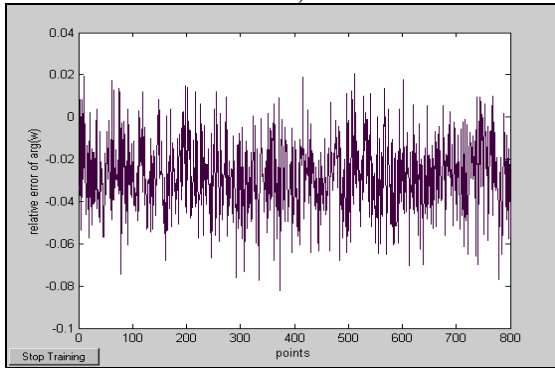


b)

Fig. 2 Relative mean error of  $|w|$ , a) for  $\max_{err}=0.1^\circ$ ;  
 b) for  $\max_{err}=0.5^\circ$



a)



b)

Fig.3 Relative mean error of  $\arg(w)$ , a) for  $\max_{err}=0.1^\circ$ ;  
 b) for  $\max_{err}=0.5^\circ$

Now let observe the blind beamforming. As previously mentioned we must use more training samples and because of high nonlinearity of the problem we should use larger number of neurons in the hidden layer. For our example we have used 80 neurons in the hidden layer and the number of training samples was doubled, compared to the DOA based beamforming. Fig. 4 and Fig. 5 give the results gained for the relative error of the absolute value of the antenna element weights and its argument. It is obvious that the results are not satisfactory. We should mention that the main problem while training was to gain low training *mse* that is a direct consequence of the high nonlinearity: the NN has a difficulty to “understand” the mapping. Namely, the NN has to learn the nature of the noise, the distribution of the information bits, and the direction of the arrivals of the signals. This leads us to a conclusion that the number of training samples must be much larger, but limitation for that exists.

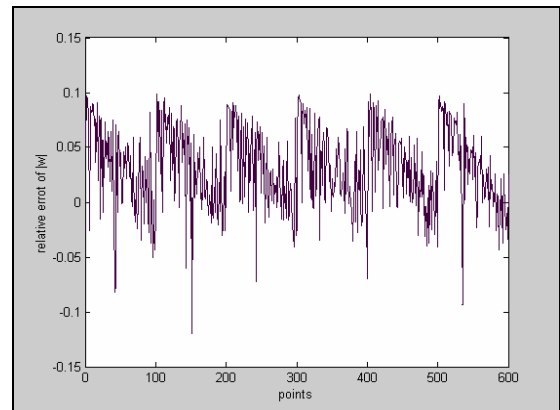


Fig. 4 Relative mean error of  $|w|$  for blind beamforming

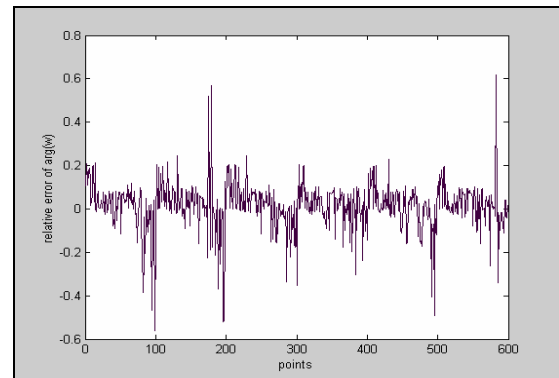


Fig. 5 Relative mean error of  $\arg(w)$  for blind beamforming

## VII. CONCLUSION AND FUTURE WORK

The problem of Null-Steering beamforming using NN approach for antenna array system was presented. Two cases were exposed. First, unlike the other authors, the estimated DOAs were used for antenna array weights NN-based determination and the imprecise DOAs estimations are taken into account. Second, the blind null-steering beamforming was presented. In this case the antenna array outputs are presented at the input of the NN without DOAs estimation.

The time needed for training the NN beamformer using the DOAs is much larger since we need to train NNs in three stages: signal detection, DOA estimation, and beamforming (Fig. 1). Also we should take into account imprecise DOAs estimations. On the other hand observing the high nonlinear relation between the  $\mathbf{R}$  and antenna element weights for blind beamforming it is a questionable is the NN able to perform the input-output mapping successfully. It is obvious that for blind beamforming we need much more training samples.

The results of computer simulations showed much better relative error performances of the first NN approach compared to the NN-based blind BF even with imperfect DOA estimations. Theoretically the RBFNN should successfully perform the blind BF and the performances might be improved by modification of the RBFNN or of the learning algorithm. Another possibility is to solve the limitations of RBFNN [8] and to provide the possibility much more training samples to be used. However these issues should be further investigated.

#### REFERENCES

- [1] Yu Hen Hu, and Jenq-Neng Hwang, "Handbook of Neural Network Signal Processing", CRC Press LLC, Boca Raton Florida, 2001.
- [2] M. M. Gupta, L. Jin, and N. Homma, "Static and Dynamic Neural Networks", A John Wiley & Sons, Inc., Publication, 2003.
- [3] C.G. Christodoulou, and M. Georgiopoulos, "Applications of Neural networks in Electromagnetics", Artech House, Inc., 2001.
- [4] R.O. Schmidt, "Multiple emitter location and signal parameter estimation", *IEEE Trans. Antennas propagat.*, vol.34, pp. 276-280, March 1986.
- [5] A.H. El Zooghby, C.G. Christodoulou, and M. Georgiopoulos, "A neural network-based smart antenna for multiple source tracking", *IEEE Trans. Antenna Propagat.*, vol. 48, no.5, May 2000.
- [6] T.J. Moody and C.J. Darken, "Fast learning in networks of locally tuned processing units", *Neural Computat.*, vol.1, pp. 281-294, 1989.
- [7] M. Sarevska, B. Milovanović, and Z. Stanković, "Alternative signal detection for neural network-based smart antenna", *Neurel'04*, Belgrade, Sept. 2004.
- [8] Maja Sarevska, Liljana Gavrilovska "Capabilities of neural network – based smart antenna for DOA estimation", *Conf. ETAI'05*, Ohrid, R.of Macedonia, September 2005.
- [9] Maja Sarevska, Bratislav Milovanovic, and Zoran Stankovic, "Reliability of radial basis function – neural network smart antenna", *Conf.WSEAS'05 on Communications*, Athens, Greece, July 2005.
- [10] Maja Sarevska, Abdel-Badeeh M. Salem, "Neural network-based null-steering beam former for smart antenna", *Egyptian Computer Science Journal*, Vol. 28, No. 1, January 2006.