

Research Article

Adaptive Beamforming Based on Artificial Neural Networks

Rajaa Amellal*

Physic Department, Laboratory Electronic Systems, Information Processing, Mechanic and Energetic, Ibn Tofail University, Kenitra, Morocco

Abstract

Adaptive beamforming is technique of signal processing play important role to increasing capacity of the wireless communication and radar systems by configured the steerable of radiation pattern and maximize gain and directivity in a direction of arrival (DoA) of desired users in order to minimizing side lobe and reducing signal to interference. We review recently the classic technique of adaptive algorithms; we specified tow method for this preprocessing beam former LMS and RLS. The least Mean Square (LMS) operate the weight vectors of antenna array elements for beamforming by iterative process as well need to be continuously adapted to the ever-changing environment. Moreover recursive least square (RLS) give advantage for fast convergence beamforming. In this paper we proved the performance of this algorithms by updating the weights in addition process based on estimated vectors using neural network, The first phase for smart beam former are used by direction of arrival (DoA) estimated using radial basis neural network (RBFNN). In next step the targets is generated from the optimum weight calculated using Minimum Variance Distortion less method (MVDLM). Finally, the simulation result for the new process is synthetized and shows using Matlab application.

Keywords

Adaptive Algorithms, Smart Beam Former, Artificial Neural Network (NN), Optimum Weights, Direction of Arrival (DoA)

1. Introduction

Adaptive beamforming is one of the adequate techniques of signal processing to develop the performance of the wireless communication and radar systems by improving the gain of main lobe in a direction of arrival (DOA) and null generation the interference in order to minimizing mutual coupling effect and maximize the directivity of antenna element [1].

Adaptive beamforming and direction of arrival (DOA) estimation constricted important areas of research in engineering telecom systems, which attired the attention of research community to give solutions for the problems impacted and increased capacity of fast communication systems [1]. Smart antennas, which are essentially adaptive array antennas cou-

pled with intelligent signal processing have emerged as an important category of adaptive algorithm capable of providing solution to this problem. Neural networks are also being widely used for developing intelligent systems and enhancing signal processing to adapted in various changes in wave propagation and electromagnetic conditions in order to upgrade signal to noise/interference ratio (SINR) [2]. Where a NN performs weights vectors for adaptive algorithms with steering location of the directions of the signals, the autocorrelation matrix invers proportionality to the element weights and thus it makes training more difficult [3], so in this paper we have preferred to estimate DoAs of incoming signals.

*Corresponding author: amellalrajae@gmail.com (Rajaa Amellal)

Received: 15 August 2023; **Accepted:** 15 September 2023; **Published:** 20 February 2024



Copyright: © The Author(s), 2024. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

User's signals arrived in antenna array modeled by matrix equation of output vectors was multiplied by complex weight vectors generated by the smart adaptive algorithm to adapt the magnitude and phase of the received signals in order to direct narrow beam patterns in the direction of desired users and nulling toward the interferer sources [4].

We presented this section will briefly explain the mean idea of programming adaptive beamforming using artificial neural network.

As explained in previous sections [3], one way to perform beamforming is to estimate DOAs of users and then according to reinforce self-learning in adaptive algorithm to estimate the antenna element weights and minimizing error [2] as well to attached desired signal. This approach is realized in two steps: DOAs estimation and beamforming.

New method based on artificial neural network training by supervised algorithm learning arranged in one procedure for update the weights vectors is presented in this paper considers the problem of Steering the main lobe of desired angle and minimizing side lobes in radiation pattern in order to check the validity of the technique for upgraded directivity of antenna [2-6].

We unified the adaptive algorithms and deep learning in

one structure, the architecture is used as a beam former in order to produce and update proper complex weights for the block diagram to signal processing. The key of this technology is to accurate direction of transmitters and steering receivers in desired users is not achieved in hardware, but rather by applying a calibration process based on deep weighted for adaptive digital beamforming.

Two cases are presented First the estimated Direction of Arrivals (DOAs) next we achieve optimization complex weights, methodologies are based on the mapping between the signal autocorrelation matrix and the angles of arrival. In the Second phase radial basis neural network (RBFNN) and FFNN (feed forward neural network) analyzed receives input vectors as combinations of DOAs estimated and produces the complex weights for the ad block processing and beamforming using adaptive algorithms RLS and LMS [4].

In digital beam forming, the operations of phase shifting and amplitude scaling for each antenna element and summation for receiving are done digitally in other way that eliminate the interferences, this is realized by varying the weights of each of the sensors (antennas) used in the array.

We need to train steps of learning in two steps: DOA estimation, and beamforming.

2. Adaptive Beam Forming Antenna Arrays System

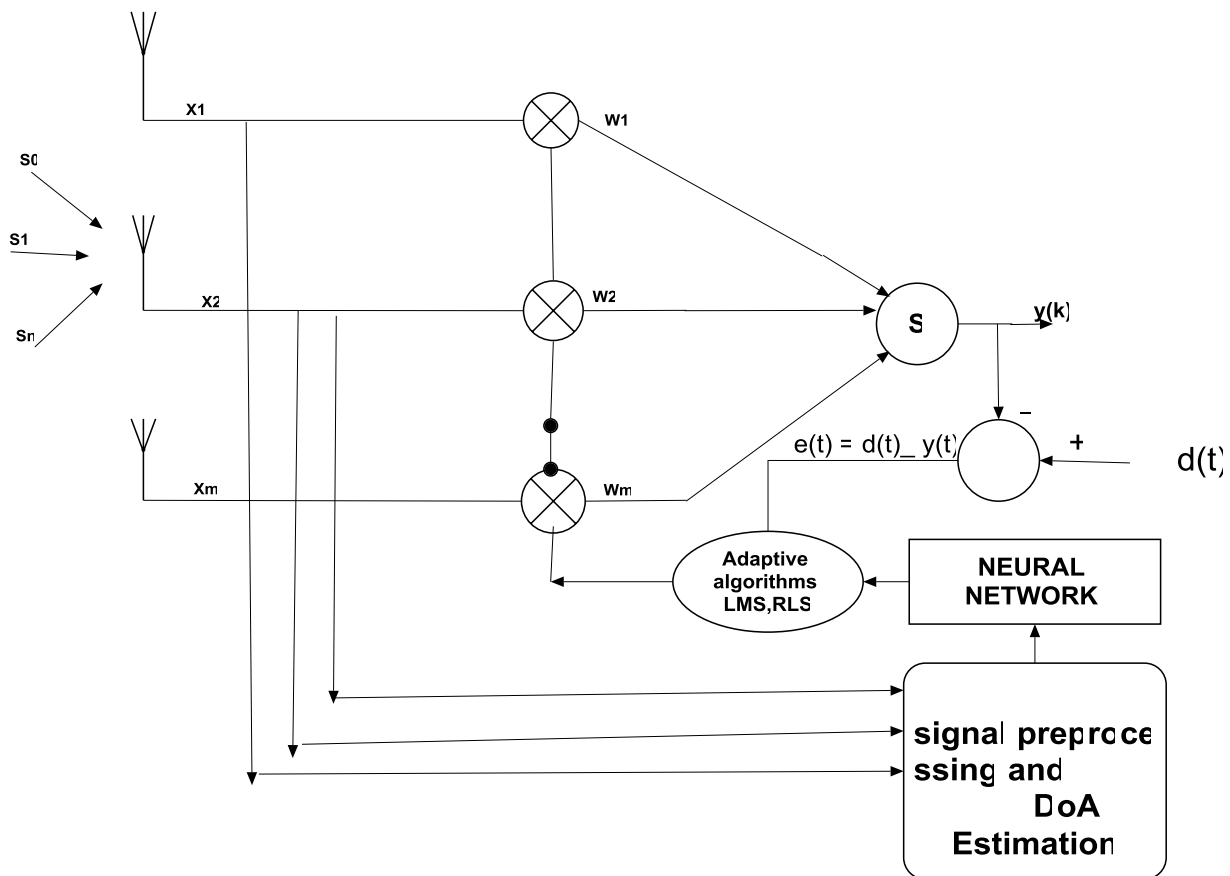


Figure 1. Block diagram of smart Adaptive beamforming (SAB).

The smart digital beam former architecture consists of antenna in users received signal preprocessing, an artificial neural network and output post processing for DoA estimation. Phase preprocessing exploits antenna expertise about the received puissance and path loss [3]. It removes undesired users, give exact values of normalized vector concerning the real part of covariance matrix signal in the input layers, and reduces the inputs to a small set of triaged information. Network preprocessing can be trained if inputs vectors are illustrated by cooperative process based in autocorrelation R_{xx} of received signal and noise produced are available to produce DoA estimation and operate the optimum weights [1].

The antenna array can be considered to be a mapping from the space of the angle of arrived target range: $\{\theta = [\theta_1, \theta_2, \dots, \theta_K]^k\}$ to the space of matrix equation in real time $\{X(t)\}$ for K users signals. The training network can successfully perform this inverse prediction for multiple source tracking [4].

Principal phase of smart beam former process attached when the neural network will estimate the weights of the new beamforming radiation pattern according of adaptive algorithms generate the outputs after the estimation of direction of arrival (DoA) when the desired signal are operated and reference signal appreciate the performance.

3. Model of Uniform Linear Array and NNs Estimation DOA

In addition process system received M users signals that are incident on the linear array antenna from different directions $(\theta_0, \theta_1, \dots, \theta_{M-1})$ [2] inclined of wave front, is given by the following equation:

$$x_n(t) = \sum_{i=0}^{M-1} S_i(t) e^{j\left(-\frac{(N-1)}{2} + n\right)kd \sin(\theta_i)} + n_n(t) \quad (1)$$

Where: $S_i(t)$ are the steering vector of received signal from multiple users, $n_n(t)$ is the additive whit Gaussian noise (AWGN)

Where the antenna elements spacing by d and k represent wave vector: $k = \frac{2\pi}{\lambda}$

The matrices form of the users signal have written as:

$$X(k) = AS(k) + N(k) \quad (2)$$

$$A = [a(\theta_1) a(\theta_2) \dots a(\theta_K)] \quad (3)$$

And $a(\theta_m)$ operate the steering vector associated with Direction θ_m and K used signals

$$a(\theta_m) = [1 e^{-jKd} e^{-j2Kd} \dots e^{-j(M-1)Kd}]^T \quad (4)$$

In (2) A is the $M \times K$ steering matrix of the array toward the direction of the incoming signals:

Autocorrelation matrix R_{xx} required of the received signals by M elements of antenna array system are generated as follows:

$$R_{xx} = \begin{bmatrix} R11 & R12 & \dots & R1M \\ R21 & R22 & \dots & R2M \\ \vdots & \vdots & \ddots & \vdots \\ RM1 & RM2 & \dots & RMM \end{bmatrix} \quad (5)$$

Vector present the input of neural network have formed by the triangle part of correlation matrix in specific first row since it contain adequate information and real part for the received signal equation as:

$z = [R11 \ R12 \dots R1M]$, then the input vector is normalized to be more suitable as input of neural network:

$$b = \frac{z}{\|z\|}$$

Optimization phase considered a suitable estimate of weights had been formed using preprocessing step based on MVDL beam former [2] is given in equation:

$$W_{op} = \frac{A_d R_{xx}^{-1}}{A_d^H R_{xx}^{-1} A_d} \quad (6)$$

Where A_d represent the steering vector of K desired signals received by antenna array with M elements.

Where the steered vector is given by:

$$A_d = \begin{bmatrix} 1 \\ \exp\{j\beta(\sin(\theta_i) d)\} \\ \vdots \\ \exp\{j\beta(\sin(\theta_i) (M-1)d)\} \end{bmatrix}, \text{ and } \beta = \frac{2\pi}{\lambda}$$

The weights that generated from MVDL equation constrained the targets of neural network can update the vectors weights of adaptive algorithms as result generate fast smart system beamforming and steer the main beam of radiation pattern toward desired signals and nulling interferences and undesired sources as well give optimal form of signals [2, 3].

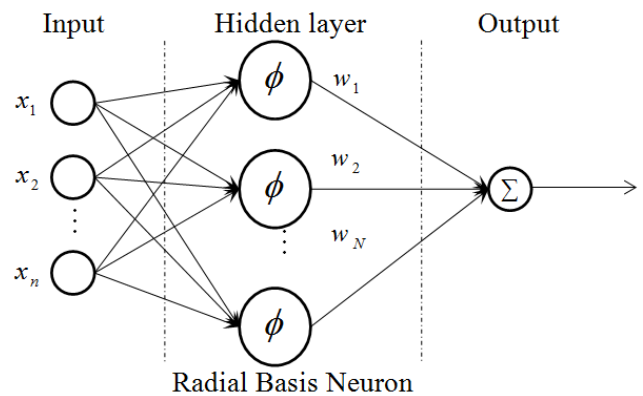


Figure 2. Radial basis neural network (RBFNN) architecture.

The RBF network architecture can improve variate mission concerning classification and prediction based on the static and variable probability for estimation using the couple (inputs, targets) such as it can be trained by supervised and unsupervised algorithm. The architecture of a one-layer RBF network consists of an input layer, a hidden layer of Gaussian method, and an output layer contained the activation function and summation nodes [5].

4. Algorithm Description

After achieving the prediction phase and desired output vectors from the artificial neural network training. The estimate weights vector have calculated from the neural network outputs from both training steps [9] (phase training and training network on desired output vector), the weight equation is represented as:

$$w(k) = y(k) * x_s^\dagger \quad (7)$$

$Y(k)$ is the neural network output obtained from the second step of training and x_s represent the specific desired signal e remaining by the angle θ'_i estimated [9] in the phase training using RBFNN.

\dagger refer to Moore-Penrose pseudoinverse of matrix

The weights vectors of NNs well memorized after reaching to best training performance and the network tested for new unknown received signals [2]. Digital Beamforming accurate the control of signal by the weight adjustment and adaptation. The purpose method center to find a set of desired DoA and optimum weights that will permit the output response of the adaptive element at each instant of iteration to converge at adequate output signal, or as fixed as possible to the desired response [7]. Moreover, the iterative process and will continue until all the weights in the array converge.

Tap-weight vector:

$$w(k) = w_R(k) + iw_I(k)$$

'R' and 'I' are denoted components real and imaginary parts, respectively.

For each step of training process: $n = 1, 2, \dots, N$ compute N user signals:

Desired response:

$$d(n) = \sum_{i=0}^{M-1} S_i(n) e^{j(-\frac{(N-1)}{2} + n)k d \sin(\theta'_i)} \quad \text{when } \theta'_i \text{ represent the estimate angles of desired users}$$

Reference signal: $r(n) = d(n) * x(t)$

Output vector: $(k) = \sum_{i=0}^N w(k) * x(k)$, when $w(k)$ represent the vectors weights of adaptive algorithm LMS, RLS

for k used signal in block SAB

Estimation error: $e(n) = d(n) - y(n)$

In first case we use the estimate DOAs at the inputs, this combination of the sources transmitters estimated and optimization phase by MVDR method correspond to given radiation pattern (antenna weight vector) more directivity and excited selectivity that produce unity response in desired direction, since the NNs is trained to give unity response only for some specified DOAs [4].

To achieve an updated dataset, some difference and incertitude concerning the desired DoAs and their comparative convergence from the respective actual DoAs they are applied in range targets for the block estimator based on FFNN. Firstly, all the desired DoAs must lie within the angular range label in this array in order to accurate the total radiation pattern and upgrade the reference signal.

The input vectors of both NNs are generated by the same group of L randomly generated angle vectors contained the targets of NNs $\theta_l = [\theta_1 \theta_2 \dots \theta_L]^T$ ($l = 1, \dots, L$), where θ_{ni} ($ni = 1, \dots, N$) are the DoA of N interference signals. In this phase DOA estimation, using NN concept is related to some degree of uncertainty [12]. Namely, the actual vector of DOAs: $\theta = [\theta_1, \theta_2, \dots, \theta_L]$ is presented with estimated vector $\theta' = [\theta'_1, \theta'_2, \dots, \theta'_L]^T$ where:

$$\theta'_l = \theta_l + \Delta\theta_l, \text{ for } l = 1, 2, \dots, L$$

The $\Delta\theta_l$ present receives random values of uniform antenna arrays.

The above methodology is summarized in the following steps:

Operate L angle vectors θ_l .

Construct the range target and inputs vectors (b, θ_l) for RBF network and generate the estimate angle vector $\theta'_l = [\theta'_1 \theta'_2 \dots \theta'_L]$.

Produce optimum weights W_{mvdr} , that correspond to $\theta_l = [\theta_1 \theta_2 \dots \theta_L]^T$ ($l = 1, \dots, L$), by applying MVDR method respectively.

Apply feed forward training to FFN networks using the pairs (θ'_l, W_{mvdr}) , $l = 1, 2, \dots, L$.

The output layers of both NNs will extract the excitation estimate weight vectors that update the weights of adaptive algorithms LMS, RLS to converge and reinforce sensors to confirmed the direction of arrival of the signal from the transmitter and accordingly the receiver should be able to steer the beam and acquire maximum signal in specified direction. Steer the main lobe towards θ_0 (angle of desired signal), place N nulls towards θ_n ($n = 1 \dots N$) Correspond to inferences signal and decrease the side lobe.

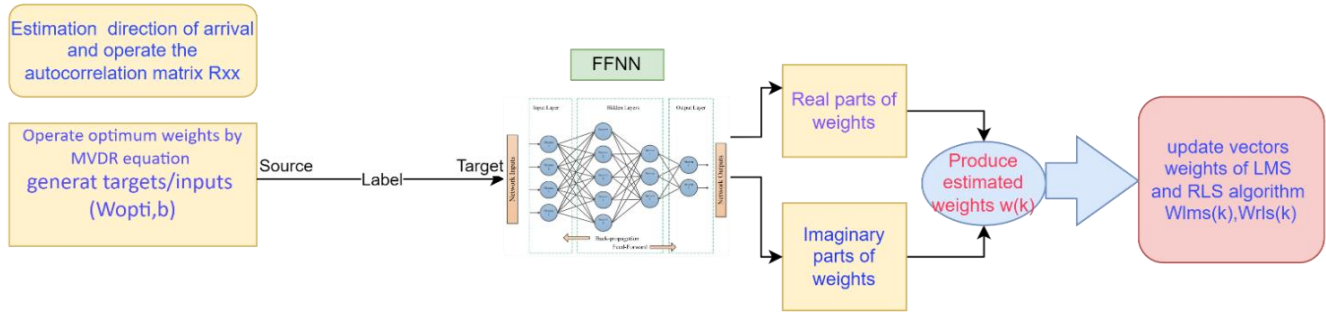


Figure 3. Diagram that illustrates the NN structure and summarizes the NN training procedure for update weights in adaptive algorithms.

5. Adaptive Algorithms LMS and RLS Based in Neural Network

Least Mean Square (LMS) method have a principal objective consist of produces and update weights multiplied by received signal This technique reduces beam width and improves directivity with high accuracy. Because it requires only the knowledge of Direction of arrival (DoAs) of incoming signals for upgrade the convergence of beamforming vectors and output signal produced as result reference signal a gained the performance location, while the MVDR technique additionally demands the calculate autocorrelation matrix of the received signal at the inputs of the array elements [10]. Some researchers have just used the normalized vector of autocorrelation matrix as input for their NNs for estimation direction of arrival on the first phase, next we can predict the new weights update this algorithm and minimizing the errors, through proper weights upgrade convergence and optimizing in both cases of neural architecture and smart antennas system [12].

This user signal as the control for the weight adjustment and adaptation. In order that the adaptation takes place a reference signal $d(n)$ must be supplied by the adaptive array. There are numerous ways to generate the reference signal [4].

The LMS algorithm avoids inverse matrix operated this method affirm to calculate autocorrelation matrix and reference signal can be calculated by following equations:

Moreover, the reference signal is:

$$r(n) = d^*(n)x(n)$$

The weight vector is found to be:

$$w(n+1) = w(n) + \mu u(n)e^*(n) \quad (8)$$

when: $e(n) = r(n) - W^H(n) * u(n)$ and $u(n)$ presented user signal of adaptive beamforming block

Here, $w(n)$ =weight vector, μ =step size and its value considered as $0 < \mu < \frac{1}{\lambda_{MAX}}$, e =error signal

Where $w(n) = w(k)$ the estimate weights calculated using NNs

LMS are highly diverse in this algorithm the input signal is defined as $X(n)$ are multiplied by the input weights $w(n+1)$, which in this case are complex [4]. The output signal is the weighted sum by the vectors weights updated using artificial neural network [11]. The output signal have given as expression:

$$y(k) = W^H(k) \cdot \bar{X}(k) \quad (9)$$

When:

$$W(k) = w(n+1)$$

In order that the adaptation takes place a reference signal $r(n)$ must be supplied by the adaptive array.

Algorithm RLS is another technique that according for minimizing error and guaranties the fast convergence of beamforming. Compared with LMS, wish make use of sample blocks to optimize R_{xx} then invert is to calculate the vectors weights, the RLS technique updates directly R_{xx}^{-1} every time an input signal sample is label, reducing the complexity of the whole process [13]. The RLS technique makes use of a parameter a called “forgetting factor” α in order to limit the effect of older samples.

$$R_{xx}^{-1}(n) = \frac{n}{(n-1)} [\alpha^{-1} R_{xx}^{-1}(n-1) - \alpha^{-1} h(n) u^H(n) R_{xx}^{-1}(n-1)] \quad (10)$$

For $n = 2, \dots, N$

Where K represents the total number of samples used by the technique to converge, and $h(n)$ is defined as:

$$h(n) = \frac{\alpha^{-1} R_{xx}^{-1}(n-1) u(n)}{1 + \alpha^{-1} u^H(n) R_{xx}^{-1}(n-1) u(n)} \quad (11)$$

The autocorrelation matrix are initialized by:

$$R_{xx}^{-1} = \delta I_{mm}$$

When δ is a large probability number, and I_{mm} is the identity matrix

The weights are recursively updated according to the expression:

$$W_{rls}(n) = W(k) + h(n)[r^*(n) - u^H(n)W_{rls}(n-1)] \quad (12)$$

Where $u(n)$ is the input signal and the H denotes Hermitian (complex conjugate) transpose. And $W(k)$ when $W(k)$ is the weight vector estimated by FFNN.

In this process, the weights $W_{rls}(n-1)$ are respectively initialized as output generating using neural network based in the result given by estimation direction of arrival signals to antenna arrays [1-3].

In this study, we present a new approach to design LMS and RLS algorithms predictors. This approach exploits the concept of deep neural networks and their supremacy in terms of performance and accuracy prediction.

Estimate the coefficient weights in order to upgrade convergence and optimizing in both cases of neural architecture and smart beam former system. Where the prediction process is realistic, the complex weights was defined as the combination of relative amplitude and phase shift for each antenna when the directions of the desired signals change as well the weights of the trained network can be used to produce the estimate weights needed to update the complex weights of the adaptive array in real time.

The basic method to develop this technology is remise temporal block of the time required by each beam former to extract the proper feeding weights [7]. The NN beam former adjusts the excitation of the adjustable weights progressively, in order to produce the most suitable radiation pattern for each new situation. After achieving the output from the deep adaptive algorithms, pointing the beam in the direction of the user. Fixed beamforming is applied to fixed arrival angle sources by the equation of array factor [2] The Array Factor (AF) is a characteristic of the weights, if we presume all elements radiate equally for every direction, direction of desired signals used with inside the antenna array that given as follows:

$$AF = AF' + |W(n-1)e - j(i-1)\alpha(k)| \quad (13)$$

Where $\alpha(k)$ is the searching angle corresponding of ue $\alpha(k) = \frac{2\pi}{\lambda} * d * \sin(\theta_i)$, when θ_i represent incidence angle in this phase DOA estimation and $W(n-1)$ weight generated using adaptive algorithm based on NNs and AF' array factor for one element antenna.

Driven by the above motivations, the adaptive Beam former Map framework is illustrated in Figure 2.

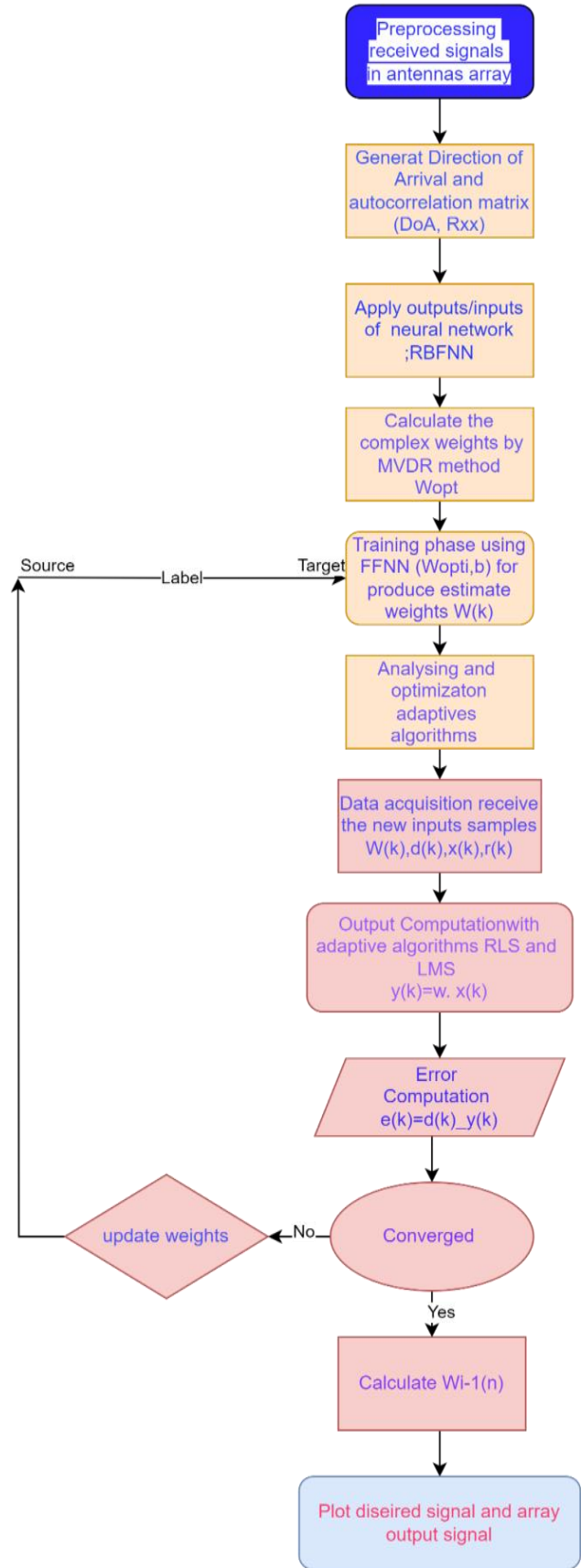


Figure 4. Flowchart of the proposed adaptive algorithms based on deep neural network.

The application of every smart adaptive beamforming (SAB) technique update complex-valued weights and cooperate the operations of phase shifting and amplitude scaling for each antenna element according by the result of estimator of DoA finally summation for receiving are done digitally. The proposed algorithm also results in complex-valued weights, because the algorithm employs an SAB technique training by the MVDR method and NNs in each iteration [1, 2] affect the convergence of preference solutions and adapt the hybridization process between NNs and adaptive algorithm as result we can proving the self-learning for beamforming system. Therefore, a mean reason to rend this method applicable arises to provide the feasibility of complex weights in reality. If the antenna array, utilized by the SAB technique concerning the proposed algorithm, operates in reception mode, then the implementation of complex weights and draw of the respective radiation pattern thus meeting after demonstrate capability of NN to learn autocorrelation matrix information of received signals, this stage are easy tasks, because they are just algorithmic procedures [10]. Results and simulations in the next section will show much better relative error performances of the DOAs estimation compiled for adaptive beamforming could potentially improve the formation of radiation pattern.

The mean response time is the mean value of the time attached by each beam former to calculate the predict weights. The smart beam former amplify and attenuate the received power or gain of signals from each antenna element, as well adjust weights progressively and magnitude, in order to produce performant radiation pattern for each iteration in ade-

quate phase [7, 8].

6. Simulation Results

Comparative experience had been realized for tow adaptive beamforming algorithms to study the performance and convergence has discussed of classic process and the proposed method. In these simulations, L sensors arrays with distance between antennas $d=0.5\lambda$ are considered. In each method, the coefficient of vectors weights are calculated according using NNs trained through a learning process using generated input/output pairs using the function concerning the supervised algorithms: for updating the complex weights and give exacted output of adaptive algorithms LMS and RLS $y(k) = W(k)*x(k)$ in order to steer beamforming in desired users. In addition, array factors are calculated and plotted for $\alpha(k)$ is the searching angle between $-90^\circ \leq \theta \leq 90^\circ$, For two dimensions we consider also the elevation angle $-90^\circ \leq \phi \leq 90^\circ$

6.1. Analysis of the LMS Algorithm

In premium step, we study the performance LMS beam former; the simulated results are shown below. Figure 5 and the normalized radiation pattern in uniform linear arrays (ULA) form respectively, the user's signals in 4-antenna array combined to form a single output. In alternative model, this is accordant by to get the direction of the desired signal by varying step-size and by setting initial weights as result decreasing the LMS mean error (ME) between the actual array output and desired signal:

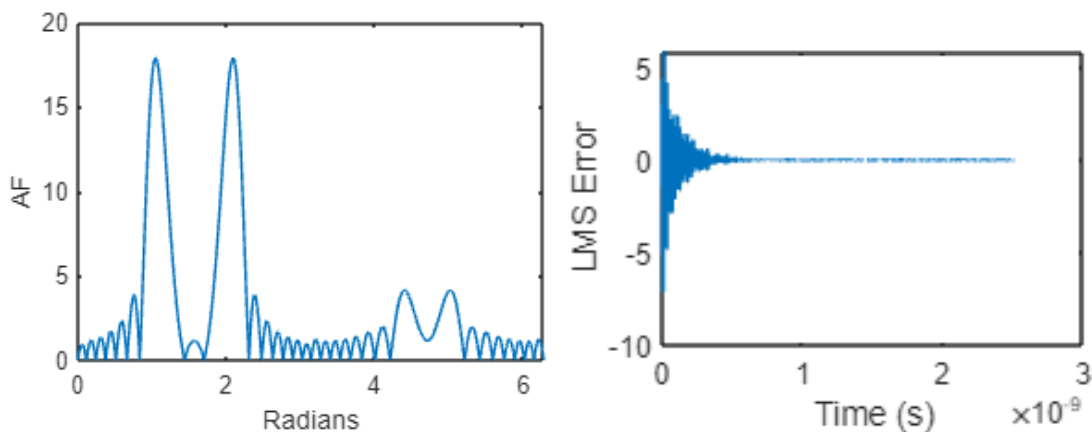


Figure 5. Normalized radiation for uniform linear arrays (ULA) and mean error (ME) for adaptive beam former LMS.

After applying the FFNN and long short-term memory (LSTM) using inputs and targets (θ'_i, W_{MVDR}) , and the estimator process of DoA and weight have been assumed for the next adaptive block system, the network had been tested for some new unseen data correspond of the received signals and

has given fast convergence to generate radiation pattern.

The linear plot of radiation pattern (uniform Array factor (AF) with respect to Angle Of Arrival (AOA)) of new incident signal with desired signal in three DOA $= \{-30^\circ, 0^\circ, 30^\circ\}$, As shown in Figure 9.

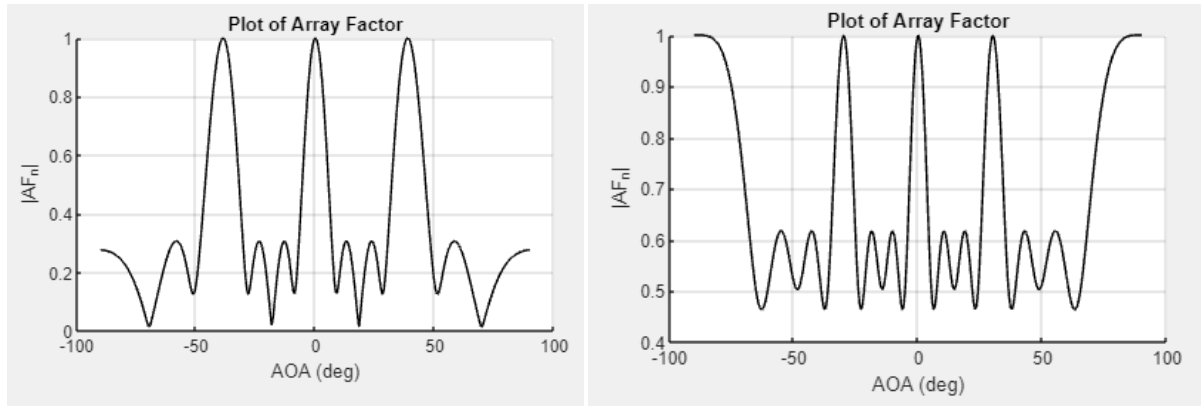


Figure 6. Antenna beam radiation pattern using FFNN and LSTM and regression ($M=4$, $d=\lambda/2$).

The high-speed capacity for learning abilities of NN are performed by calculate the weight and bias values based on the LM training algorithm.

The smart adaptive process it can be demonstrated, while the measured mean square error MSE between reference and predicted signal and loss function measure incertitude of ANN are again due to decrease in the learning rate caused by the proposed method. At the end of the training process after 100 epochs, the test MSE has significantly had been reduced according by variate the number of nodes of NNs, as shown in Figure 7.

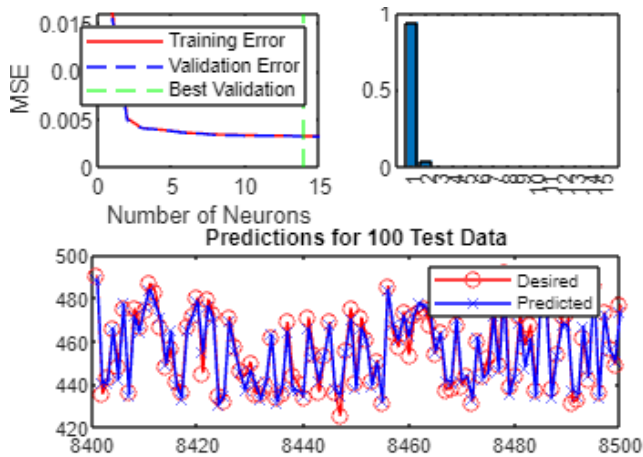


Figure 7. Learning curves optimization of the LSTM and regression and mean square error (MSE) between the actual array output and desired output.

In other way for this application, desired sources are located at azimuth and elevation angel's θ and ϕ ranging from -90° to 90° to span the field of view of the antenna. Once the ANN is trained with a representative set of training input/output pairs we used the test pattern and measured the time required for the 2D RBFNN-DOA estimation. In the performance phase, the NN produces estimation of the

weight vector $W(n)$, and the radiation pattern is show in Figure 8:

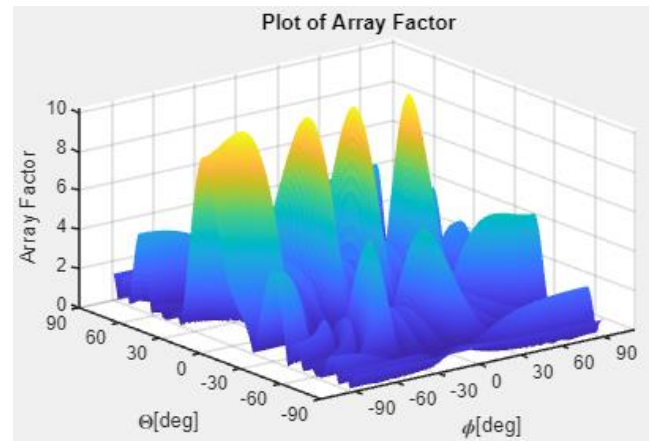


Figure 8. Antenna beam pattern using LSTM and regression weighted LMS beam former for azimuth angle θ and elevation angle ϕ varied in the range $DOA = [-90^\circ, 90^\circ]$.

6.2. Analysis of the RLS Algorithm

RLS is another technique that mean objective is decrease the mean square error (Mse) and the beam former is accurate to produce a radiation pattern where a side lobe will generated at a small angle incertitude $\Delta\theta_i$. We selecting number of elements= 4, a number of snapshots=200, the spacing between the array elements, $d=\lambda/2$, the value of forgetting factor $\alpha = 0.91$, AOA of the interference signal= -60° and AOA of desired signal $=30^\circ$. The normalized array pattern formed by RLS algorithm is shown in Figure 9 after applying the artificial network for update and weighted this RLS beam former we can observed the main lobe are steered in new desired angle of arrival $DOA = 0^\circ$ and side lobe will minimized.

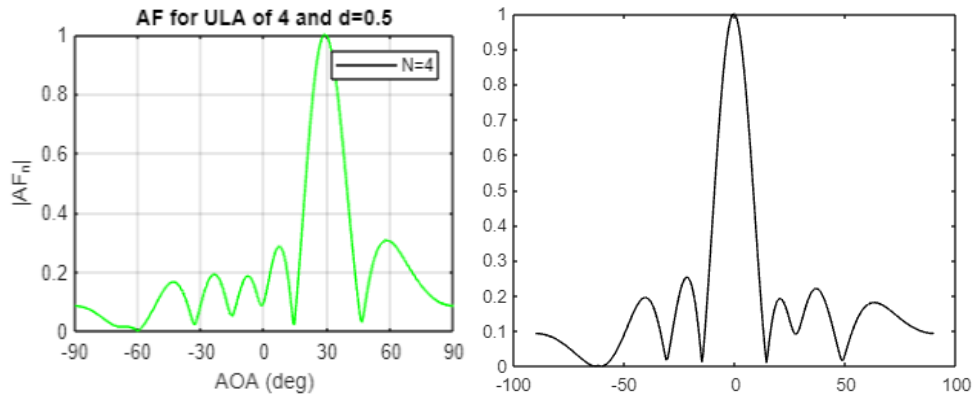


Figure 9. a/c Normalized array pattern with ($M= 4$ and $d= \lambda/2$), beam radiation pattern using RNN.

First step of the FFNN training, input data are produced to the RBFNN in prediction phase of DoA estimation and targets configured by optimum weights calculated using MVDR method, next the Adam optimization algorithm is used to update the hidden layers, while the MSE metric shown in (Figure 11) is used to demonstrate the validation result of FFN network. Given that every hidden state and consequently the output of the RBFNN will have a different size, an extra linear transformation layer have placed after the output layer. Next, the LSTM-RNN and FFNN based on (θ'_i, W_{MVDR}) is tested in terms of deep learning and target

values of outputs vectors must be excited in order to calculate the predict weights and decreasing cost function for updated and weighted RLS beam former, a statistical analysis of the results derived from this test is given complex weights estimated in Table 1. It seems that the LSTM and recurrent neural network (RNN) implementation provides best accordance in magnitude and regarding peak of the main lobe on high gain and the side nulls placement, an example of radiation patterns of normalized Array factor (AF) with respect to Angle Of Arrival (AOAs) shown in figure 10.

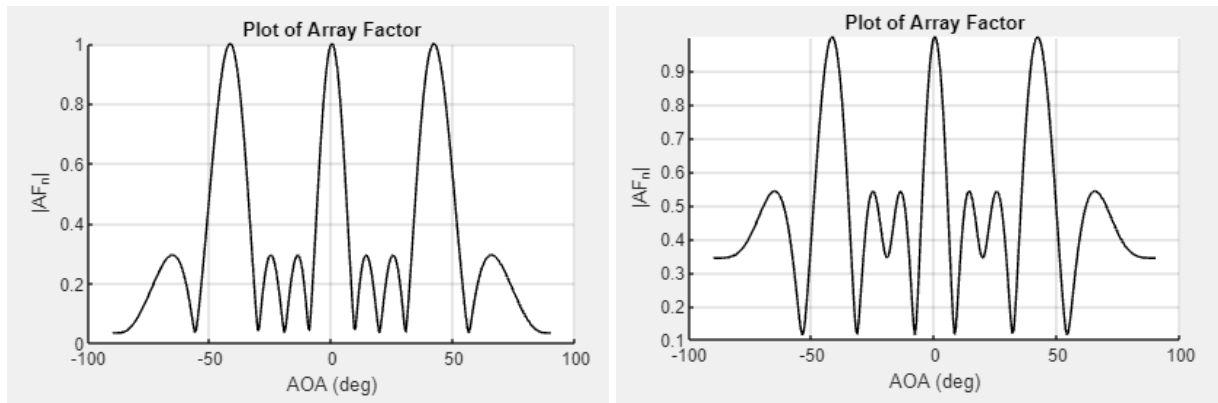


Figure 10. beam patterns produced by RLS beam former weighted using FFNN and LSTM for Interference angle received at 50° and AoAs equal to 30° and $-30^\circ, 0^\circ$.

After the FFNN, training phase based on LM (levenberg-marquardt) algorithm has operated to update weight for RLS algorithm; the NNs have been compiled from some new received signals and has given good performance.

Table 1. The optimum complex weights in the case for which the algorithm converges.

Algorithms	Weights
LMS algorithm weighted by FFN network	$W1 = -0.685790925080265 + 0.614041620136966i$
	$W2 = 0.0253686329620717 + 0.920680928099831i$
	$W3 = 0.645860820010554 - 0.764173279516116i$

Algorithms	Weights
LMS algorithm weighted by LSTM-regression	$W_1 = 0.0283874664455652 + 0.862008333206177i$ $W_2 = 1.74389898777008 + 0.725007116794586i$ $W_3 = 1 + 3.40776077695826e-17i$
RLS algorithm weighted by FFNN	$W_1 = 0.53421797034364 - 1.26721604531722i$ $W_2 = -1.36849305736739 - 0.0544553019757699i$ $W_3 = 0.359745469079543 + 0.93810837365962i$
LSTM-regression weighted RLS algorithm	$W_1 = 1;$ $W_2 = 1.630662757079 - 1.03738223737032i$ $W_3 = 0.424563872889803 - 0.905515000047208i$

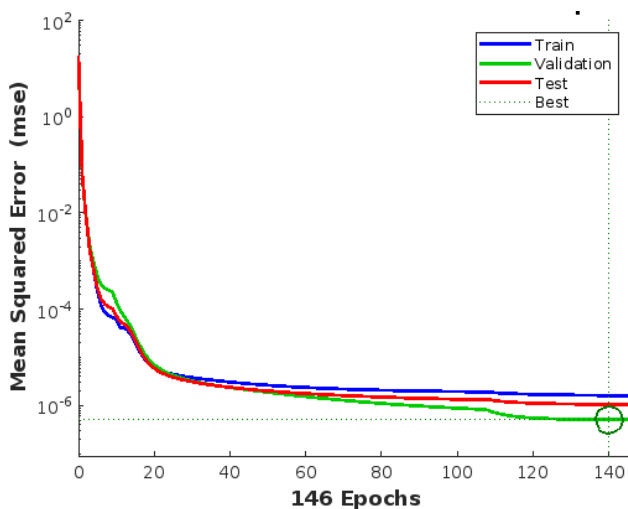


Figure 11. Performance correspond the mean square error (MSE) of FFNN.

7. Conclusion

Smart adaptive beamforming according by artificial neural network guaranties a reasonable performance for tracking the desired signal while simultaneously nulling the interference sources in the process based in adaptive algorithms LMS and RLS exposed by two cases DOA estimation and prediction weights. This method is much less sensitive to some types of mismatches and the small training sample size than the other algorithms. Furthermore, the proposed beam former based on weights estimated and optimized computed using a FFNN trained by two different back propagation-learning algorithms. The simulation result was implemented for $M=4$, $d=\lambda/2$ and showed that the performance of Levenberg-Marquardt (LM) training is the best.

The ABF is just one of the two processes performed by a smart antenna to control the reception of incoming signals. The integration of a DoA estimation process into the current RBFNN model will significantly enhance the operation of smart antennas and minimizing error in order to upgrade the

convergence of adaptive algorithms and moderate signal to noise (SNR) conditions. In this paper we applicate the proposed method for 2D antenna arrays it was considered good practice to approach the new ABF problem in its simplest form and optimum complex weights was calculated. As a future Enhancements for this method it is possible to increase the number of desired users (targets) and reinforce the DoA estimation phase to the update the reference signal for the LMS and RLS algorithm in order to realize the auto-adaptive beamforming system then the artificial neural network give advantage the faster and convergence to produce radiation pattern. As result as using lower complexity ways in the hybridization process between the artificial neural network and adaptive algorithms.

References

- [1] Rawat, Abhishek Yadav, R. N. Shrivastava, S. C.(2012), "Neural network applications in smart antenna arrays: A review", AEU - International Journal of Electronics and Communications Volume 66, Issue 11, Pages 903-912.
- [2] H., AdheedAhmed, Sulaiman. (2015), International Journal of Computer Applications "Elman Recurrent Neural Network Application in Adaptive Beamforming of Smart Antenna System", Volume 129, Issue 11, Pages 38-43.
- [3] Mallioras, Ioannis Zaharis, Zaharias D. Lazaridis, Pavlos I. Pantelopoulos, Stelios. (2022), IEEE Transactions on Antennas and Propagation "A Novel Realistic Approach of Adaptive Beamforming Based on Deep Neural Networks", Volume 70, Issue 10. Pages 8833-8848.
- [4] Liu, Zhang Meng, Zhang, Chenwei, Yu, Philip S. (2018), "Direction-of-Arrival Estimation Based on Deep Neural Networks with Robustness to Array Imperfections" IEEE Transactions on Antennas and Propagation, Volume 66, Issue 12, Pages 7315-7327.
- [5] Potocnik. (2012), "Neural Networks: MATLAB examples" Mathworks, Pages 1-91.
- [6] Toscano et al. (1999), "G () = e + (1 0) e 0", Society Volume 46, Issue 12, Pages 1997-1999.

-
- [7] Saffah & Hreshee. (2016), "LMS Algorithm for Optimizing the Phased LMS Algorithm for Optimizing the Phased Array Antenna Radiation Pattern", Volume 22, Issue 2013, Pages 1-6.
- [8] C. Liu & Helgert. (2020), "An Improved Adaptive Beamforming-based Machine Learning Method for Positioning in Massive MIMO Systems", International Journal on Advances in Internet Technology Volume 13, Issue 1, Pages 21-34.
- [9] Albanee. (2016), "Smart Antenna Adaptive Beam Forming Base on Neural Network With Different Training", Volume 20, Issue 3, Pages 60-74.
- [10] Zaharis et al. (2012), "Comparative study of neural network training applied to adaptive beamforming of antenna arrays", IEEE Transactions on Antennas and Propagation Volume 126, Issue 2014, Pages 269-283.
- [11] Mok atren et al Mokatren, Lubna Shibly, Cetin, Ahmet Enis, Ansari, Rashid. (2019), "Deep Layered LMS Predictor", Pages 1-5.
- [12] Sarevska & Salem. (2006), "Antenna Array Beamforming using Neural Network", Engineering and Technology Volume 1, Issue 7, Pages 115-119.
- [13] Dakulagi & Alagirisamy.(2020), "Adaptive Beamformers for High-Speed Mobile Communication", Wireless Personal Communications Volume 113, Issue 4, Pages 1691-1707.