We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Detect and Pointing Algorithms Performance for a 2D Adaptive Antenna Array

Tiago Varum, João N. Matos and Pedro Pinho

Additional information is available at the end of the chapter

Abstract

In recent decades, we have witnessed a great progress in wireless communications. The huge amount of data that users expect to access has required an effort to increase the capacity of wireless networks. The main limitation of these communication systems is the increasing interference between channels and multipath fading. Smart antennas technology has emerged, solving some of these problems and improving the performance of wireless networks. This chapter addresses a group of algorithms, directions of arrival (DOA) and beamforming, applied to planar antenna arrays. The algorithms are simulated, and their performance is evaluated in terms of runtime, accuracy and dependence with signal-to-noise ratio (SNR), applied to a smart antenna system.

Keywords: adaptive antenna, direction of arrival, beamforming, planar array

1. Introduction

Adaptive antennas [1–3], often referred to as smart antennas, have a great potential over all the future wireless communications and have been a topic of relevant research and development over the past years.

This type of antennas attracted the attention in the past, mainly by the military branch, although it has a greater progress in recent years, as the result of technological developments, especially in digital processors and new signal-processing techniques. Nowadays, smart antennas are attractive for several areas including military applications, software-defined radio, satellites, mobile communications (especially in base stations), 4G MIMO and the emerging 5G MIMO.

An adaptive antenna consists, in a simple way, of an antenna array and a sophisticated signal processing. It continuously adjusts the feeding of its array elements, and therefore its radiation characteristics, to adapt to the changes in the environment around them, improving the communication.

The dynamic and flexible adjustment of the radiation pattern using a relevant signal-processing capability allows to respond, in real time, to the communication constraints, first by detecting the signal directions of arrival (DOA), and then, according with them, applying beamforming techniques to shape and steer the radiation pattern of the array, by adjusting the locations of the maximums and nulls, to achieve a better reception or transmission.

In this sense, it is possible to recover a weak signal immersed in strong interference environment where signals come from various directions. Furthermore, these antennas enable the reduction of delay spread effects and a greater ability to reduce the impact of multipath fading.

Figure 1 shows the basic block diagram of an adaptive antenna, in the receiving mode, in which some of the different elements that comprise an adaptive antenna are noticeable.

This structure starts by the antenna array and the entire underlying radio frequency (RF) chain, which includes the possible frequency downconversion and analogue-to-digital conversion. Another important component is the DSP (digital signal processor) unit that handles all signal processing of the adaptive antenna.

Using samples of received signal from each array element, DOA algorithms are applied to estimate the arrival directions of the signal or signals that impinge on the antenna array. Then, using an auxiliary intelligence process, the detected signals are divided in terms of signals-of-interest (SOI) and signals-not-of-interest/interferences. Finally, beamforming algorithms are employed using this information (DOA's and SOI's/SNOI's), computing the necessary weights (W_1, W_2, …, W_n) to apply to each array element, obtaining the desired radiation pattern, pointing the antenna to directions of interest signals, and reducing the impact of the interferences.

This chapter is divided into five sections and explores the main concepts presented in the review article [1], from which some tables and figures are reprinted with permission. It starts with an introduction to adaptive antennas, and an adaptive system applied to two-dimensional (2D)
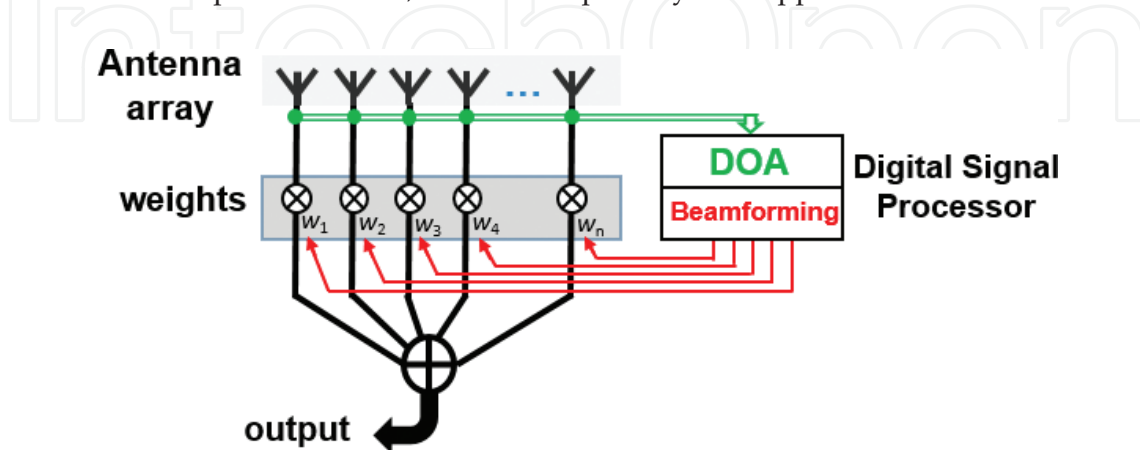

Figure 1. Adaptive antenna block diagram.

arrays is presented in Section 2. In Section 3, two DOA algorithms are presented, which also are tested and compared in terms of important factors as runtime and estimation errors. In Section 4, a group of beamforming algorithms are reported, analysed and applied to a planar 4 × 4 array. Finally, the last section summarizes the main conclusions.

2. Planar antenna array system

The system consists of planar antenna array, as illustrated in **Figure 2**, which is affected by a group of signals including SOI's and SNOI's, resulting in an input signal $x(t)$ that is composed by contributions of all these arriving signals, and noise. A sample of this signal $x(t)$ is then processed to estimate the angles of arrival of the various signals involved. Afterwards, the respective weights W to apply to each element of the array are calculated, to create a proper radiation pattern, pointing it to the direction of interest and reducing the influence of the signals from interfering directions.

Consider a planar antenna array with M × N elements, spaced by d_1 in the rows and d_2 in the columns, in which impinge J different signals $s(t)$ of frequency with wavelength λ, coming from a direction (θ, ϕ), where θ is the elevation angle and ϕ the azimuth angle. The input signal received by each element (m,n) of the array has components from each of J arriving signals and noise $n(t)$, and is given by [4, 5]

$$x_{mn} = \sum_{i=1}^{J} s_i(t) e^{j\frac{2\pi}{\lambda}[u_i d_1(m-1) + v_i d_2(n-1)]} + n(t) \tag{1}$$

with,

$$u_i = \sin\theta_i, v_i = \cos\theta_i \sin\phi_i, \ m = 1...M, \ n = 1...N, \ i = 1...J$$

The received data $x(t)$ and the noise $n(t)$ of the whole antenna array can be represented in a vector structure $X(t)$ and $N(t)$ as follows:
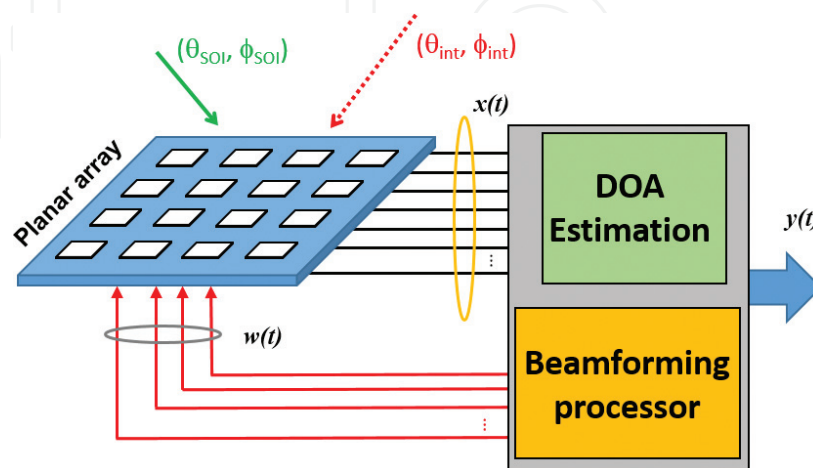

Figure 2. Planar adaptive array system.

$$X(t) = [x_{11}(t)x_{21}(t) \ldots x_{M1}(t)x_{12}(t) \ldots x_{MN}(t)]^T \tag{2}$$

$$N(t) = [n_{11}(t)n_{21}(t) \ldots n_{M1}(t)n_{12}(t) \ldots n_{MN}(t)]^T \tag{3}$$

where T denotes the matrix transpose operation.

The total input signal $X(t)$ can be also expressed by the following formula:

$$X(t) = \sum_{i=1}^{J} s_i(t) \, A_i + N(t) \tag{4}$$

where A represents the steering matrix of the planar antenna array and is given by Eq. (5). The operation $\otimes$ denotes the Kronecker product and C_u and C_v are the steering vectors in each direction of the planar array.

The steering vector [6] contains the set of phase delays that a wave will take, relating to each element of the array, and for a planar array this can be represented by Eqs. (6) and (7) [4, 7].

$$A = C_u \otimes C_v \tag{5}$$

$$C_u = \left[1 \; e^{j\frac{2\pi d_1}{\lambda}[(2-1)u_i]} \; \ldots \; e^{j\frac{2\pi d_1}{\lambda}[(M-1)u_i]}\right]^T \tag{6}$$

$$C_v = \left[1 \; e^{j\frac{2\pi d_2}{\lambda}[(2-1)v_i]} \; \ldots \; e^{j\frac{2\pi d_2}{\lambda}[(N-1)v_i]}\right]^T \tag{7}$$

The output signal of the adaptive antenna is therefore the product of the received signal $X(t)$ by the beamforming weights W, estimated through the beamforming algorithms, and applied to each array element, as the following:

$$y(t) = W^H X(t) \tag{8}$$

$$W(t) = [w_{11}(t)w_{21}(t) \ldots w_{M1}(t)w_{12}(t) \ldots w_{MN}(t)]^T \tag{9}$$

In the next sections, a system is implemented and studied using MATLAB [8]. The signal $X(t)$, which consists of the sum of a number of generated signals from several directions (θ,ϕ), being some of these signals SOI and others SNOI, adding up some noise, is created. The locations of the generated incoming signals are totally unknown beyond this step. Then, $X(t)$ is sent as a single input parameter for DOA algorithm, which provides as output parameters the estimated number of signals that reach at the array, and their locations (θ,ϕ).

After the locations of all the signals impinging at the antenna are identified, it is necessary to distinguish the signals between the SOI and interferences, and then a beamforming algorithm to determine the necessary weights to apply to each element of the array is applied.

By simulation, using high-frequency structural simulator (HFSS) [9], with the ability to change the amplitude and phase of feeding of a simulated array structure, these sets of estimated weights are applied to the array, and its radiation pattern is analysed.

3. DOA algorithms

The processor that estimates the DOAs of arriving signals to the antenna array is a crucial part of an adaptive antenna, allowing to understand the environment in which the antenna is inserted. By processing the electromagnetic waves that reach an antenna array, it is possible to extract a number of information about them, particularly their arrival directions. This processing is done by using DOA algorithms.

The arriving signals could be divided into SOIs, which are important to steer the antenna towards them, and/or interfering signals (SNOIs), whose impact on the system should be reduced.

There are three main classes of DOA estimation methods referred in the literature, differing mainly in the performance and its computational requirements [10, 11]: the classical, the maximum likelihood and the subspace methods.

The classics are based in the beamforming, in which the central idea is to scan the antenna beam over the space, and the peaks of received power are the DOAs. These methods are theoretically simple but involve a high computational effort and provide a poor performance and a low resolution. The maximum likelihood methods present higher performance than the others (especially with low SNR conditions), because they can take advantage of using better signal and noise models to provide better DOA estimation. However, due to the needs to solve nonlinear multidimensional optimization problems, it increases the required computational load, which makes these methods less popular.

Finally, there are several subspace methods for DOA estimation, which have become popular and widely studied over the last decades due to their good trade-off between the computational complexity and good performance. These methods are based on the eigen decomposition of the estimated covariance matrix of the data received by antenna array, into a signal subspace and a noise subspace. The performance of these methods is essentially limited by the accuracy of distinguishing the signal and the noise subspaces in the presence of noise.

For a planar uniform antenna array, the most applied algorithms are 2D Multiple Signal Classification (MUSIC) and the 2D Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), which are subspace-based methods, and will be subject of a more detailed description below.

3.1. 2D MUSIC algorithm

The Multiple Signal Classification (MUSIC) is perhaps the most popular DOA estimation algorithm, which assumes that the steering vectors of the incoming signals lie in signal subspace, and are orthogonal to the noise subspace. The algorithm search in all possible steering vectors that are orthogonal to the noise subspace of the covariance matrix of the received data (R_{xx}) [11–14].

The MUSIC algorithm, as illustrated in **Figure 3**, uses the received information from each element of the array, and through eigenvalue or singular value decomposition of the R_{xx} matrix, it estimates the noise subspace (U_N).
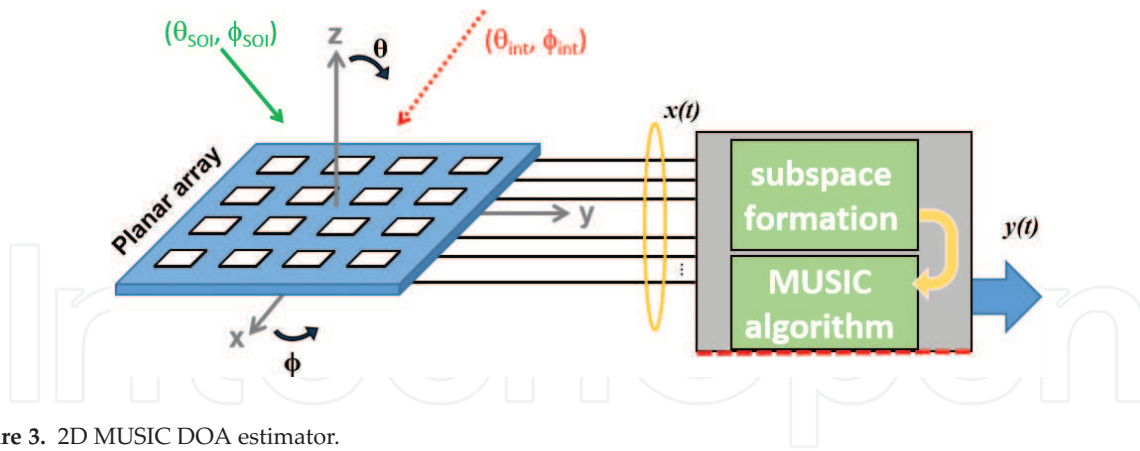
Figure 3. 2D MUSIC DOA estimator.

After the noise subspace U_N is identified, the DOAs are the resulting peaks of the MUSIC spectrum $P_{\mathrm{MUSIC}}(\theta,\phi)$, given by Eq. (10) [13]:

$$P_{\mathrm{MUSIC}}(\theta,\phi) = \frac{1}{s^H(\theta,\phi)U_N U_N^H s(\theta,\phi)} \tag{10}$$

where H represents the conjugate transpose matrix (Hermitian).

When a steering vector $s(\theta,\phi)$ is related to one arriving signal, the product of $s^H(\theta,\phi)U_N = 0$, ideally, the function assumes a high value (peak), and therefore (θ,ϕ) is the DOA. There may be several signals from different angles of arrival, creating several peaks in the MUSIC spectrum.

The MUSIC algorithm is simpler to understand and can be applied in all antenna array geometries. However, computationally it requires a lot of resources, since it has to calculate the MUSIC spectrum, Eq. (10), for all the possible steering vectors to estimate the expected peaks.

The estimation error of the MUSIC algorithm is significantly influenced by the angle grid interval in which Eq. (10) is evaluated. In the presence of coherent signals, as in multipath environments, spatial smoothing schemes [15, 16] must be applied to suppress the correlations between the incoming signals.

3.2. 2D ESPRIT algorithm

Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm is a different subspace-based DOA estimator [17–21]. This algorithm solves some of drawbacks of the MUSIC, in terms of the high computational requirements, and the resulting effects of array calibration errors. The ESPRIT algorithm does not require a high level of calibration in the array since it employs the property of shift invariance of the antenna array. Also, the computational complexity of the ESPRIT is reduced in comparison to MUSIC because it imposes some constraints on array structure.

The ESPRIT algorithm assumes that the separation between equivalent elements in each sub-array is fixed, as shown in **Figure 4**, and therefore the array presents a translational invariance.
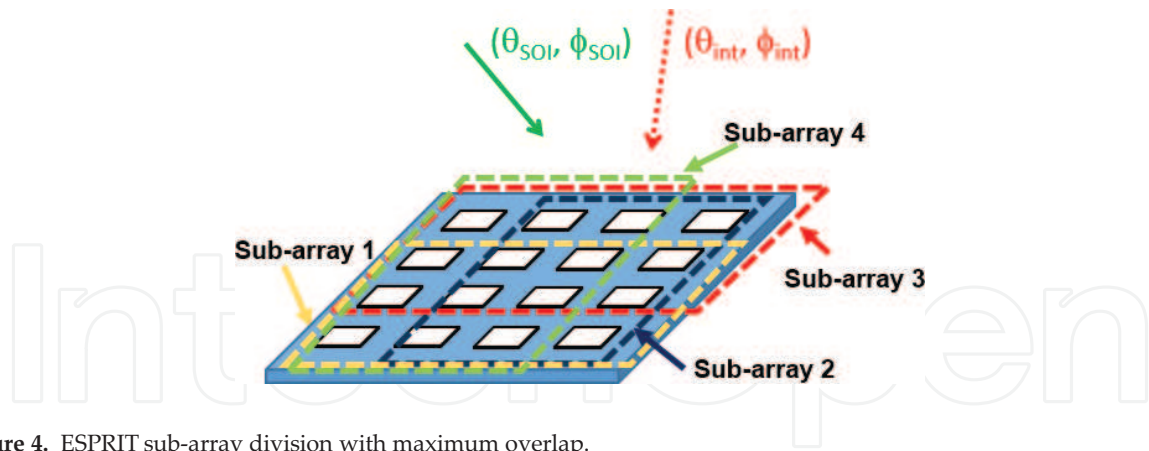
Figure 4. ESPRIT sub-array division with maximum overlap.

This translational invariance leads to a rotational invariance of the signal subspace that will enable to estimate the DOAs.

This algorithm performs three main stages: the signal subspace estimation, the solution of the invariance equation and the DOA estimation.

The algorithm procedure is [17–19] as follows:

Step 1. Signal subspace estimation

Computation of the U_s

Step 2. Solve the invariance equation

$$K_{u1}U_sY_u = K_{u2}U_s$$

$$K_{v1}U_sY_v = K_{v2}U_s$$

where K_{u1}, K_{u2}, K_{v1} and K_{v2} represent the two pairs of transformed selection matrices, while Y_u and Y_v are the real-valued matrices.

Step 3. DOA estimation

λ_i $i = 1\ldots d \rightarrow$ eigen values of $Y_u + j\, Y_v$

$u_i = 2\,\tan(Re\{\lambda_i\})$,

$v_i = 2\,\tan(Im\{\lambda_i\})$,

$\phi_i = \arg(u_i - jv_i)$ $\theta_i = \sin^{-1}(\|u_i - j\,v_i\|)$

where θ_i and ϕ_i are the DOA angular information.

3.3. Test and comparison of the DOA algorithms

A system composed by a signal generator (with signals of interest and interferences) followed by a DOA estimator and beamforming was implemented using the MATLAB [8]. The two DOA algorithms, 2D MUSIC and 2D ESPRIT, were accomplished and applied to a planar array configuration.

To test the beamforming results, a planar antenna array of 16 square microstrip elements, arranged in a 4 × 4 planar structure was designed (for a central frequency of 12 GHz), with

spacing between elements in both directions of $d = 0.5\lambda$, as shown in **Figure 5**. The array was simulated with HFSS [9] electromagnetic simulator.

Several simulations were performed with good results. As an example, two signals with directions $(\theta,\phi) = (45°, 45°)$ and $(70°, 0°)$ were generated and 'received' by the 4×4 array. With the received signal from each element of the antenna array, given by Eq. (2), the DOAs were estimated using the 2D MUSIC algorithm.

The 2D MUSIC algorithm creates a two-dimensional grid, in the range which the angles vary $\theta \in [0, 90]$ and $\phi \in [0, 360]$, and then evaluates Eq. (10) for each point of the grid.

Figure 6 illustrates the spatial graph of the MUSIC spectrum, resulting in the algorithm, which has peaks in the position of incident signals.
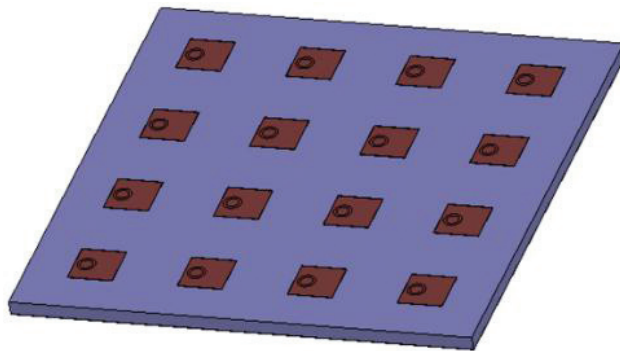

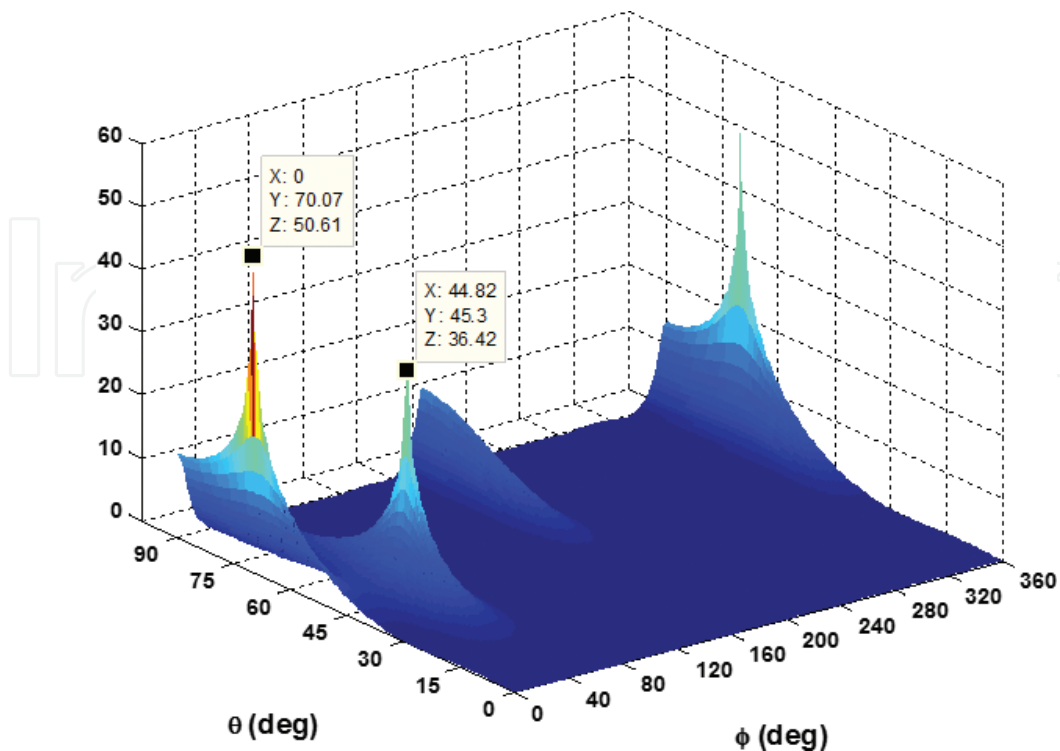**Figure 5.** Simulated planar array.

Figure 6. MUSIC spectrum.

According to **Figure 6**, the MUSIC spectrum contains two evidenced peaks. Note that there is another peak but is assumed to be repeated, since 0° and 360° are the same spatial locations.

In order to simplify the definition and extraction of the peaks of the 2D MUSIC spectrum, a function to correctly detect the number of maximum values was implemented. This function provides the points of zero gradient, and its result is shown in **Figure 7**, with the two well-defined peaks.

The output of the 2D MUSIC algorithm is that the incident signals arrive to antenna from (θ,ϕ) (45.3°, 44.82°) and (70.07°, 0°), which are quite close to the initially proposed angles.

Identical signal given to the 2D MUSIC algorithm was provided to the 2D ESPRIT algorithm, also implemented with MATLAB. The output of ESPRIT is just the pair (θ,ϕ) of estimated DOAs, since it does not have a grid to evaluate by a function. The ESPRIT algorithm was preformed and the output result estimates that the signals are arriving from (45.02°, 45.11°) and (69.82°, 0.03°).

The implemented DOA algorithms reveal estimated results very approximate to the original and expected values. These algorithms only receive the signal *X(t)*, and provide the spatial position of each incoming source that compose it.

The analysis of the performance of the 2D DOA algorithms was performed for a large number of experiments. In **Figure 8**, the runtimes of both subspace-based DOA algorithms for *n = 50* experiences are presented.

The upper part concerns the 2D MUSIC algorithm while the bottom is about the 2D ESPRIT algorithm. In both graphs, the line of the average time of all samples is identified. There is huge
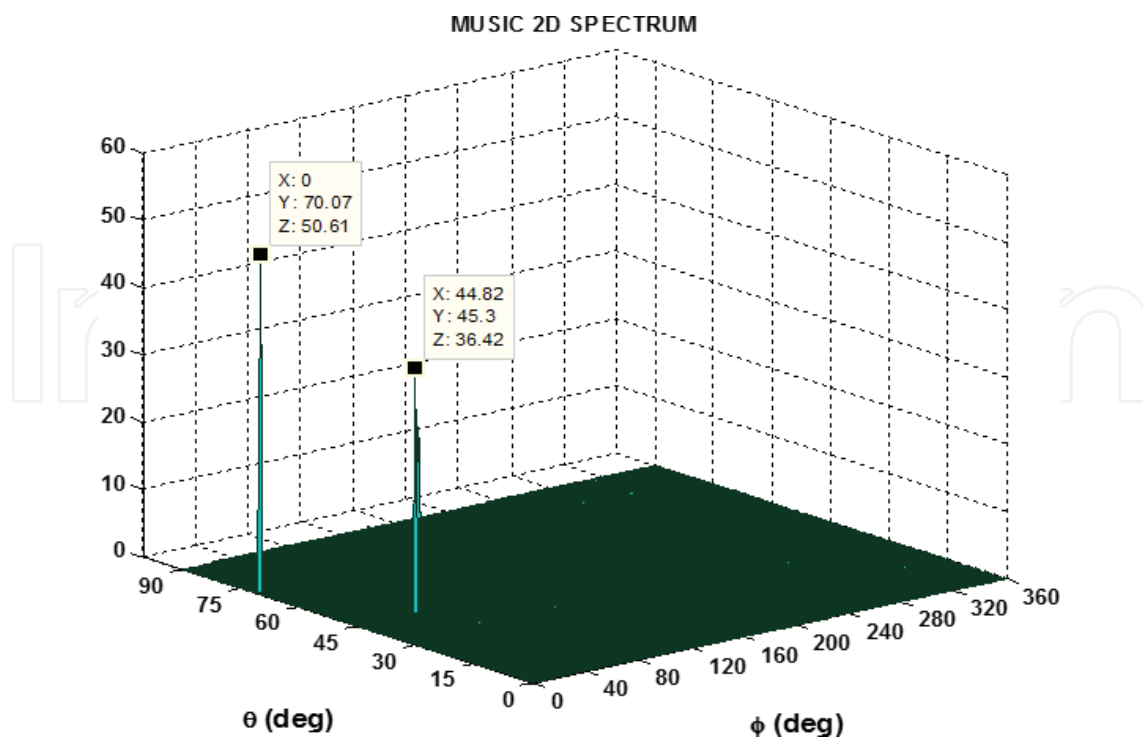

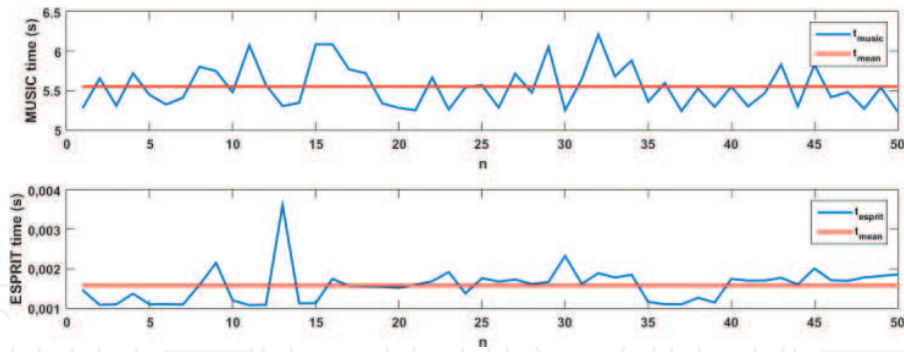**Figure 7.** 2D MUSIC spectrum peaks.

Figure 8. Runtime of DOA algorithms over *n* samples.

difference in the time that MUSIC algorithm takes to perform compared with the ESPRIT. This runtime of the MUSIC is much higher than that of ESPRIT essentially because the MUSIC algorithm needs to calculate the MUSIC function for each possible steering vector.

The 2D MUSIC algorithm takes between 5 and 6 s to estimate the DOAs with an average time of 5.4 s, while 2D ESPRIT, in the majority of samples, varies between 1 and 2 ms, with an average execution time of 1.57 ms.

Another property that can be analysed is the estimation error, between the real location of the incoming waves and the estimated DOAs from both algorithms. **Figure 9** shows the progress of the estimating error over the set of *n* experiences, using the two DOA algorithms. The upper graph is related to the coordinate θ, while the bottom is about the ϕ, and the lines of the average error are further presented.

According to **Figure 9**, it is possible to notice that the 2D MUSIC algorithm has a constant estimation error, with an average value of 0.302° in θ and 0.3052° in ϕ coordinate. The 2D ESPRIT algorithm has average errors extremely lower than 2D MUSIC, about 0.037° in θ and 0.015° in ϕ coordinates.
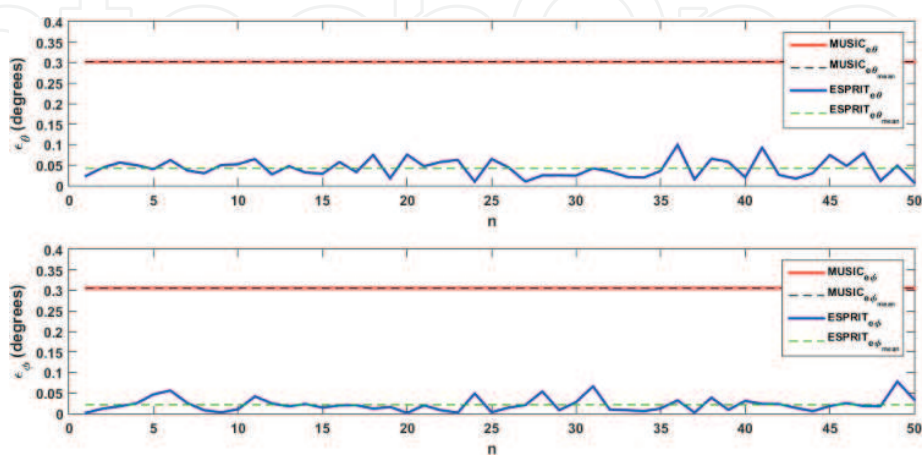

Figure 9. Estimation error in θ and ϕ coordinates for MUSIC and ESPRIT algorithms.
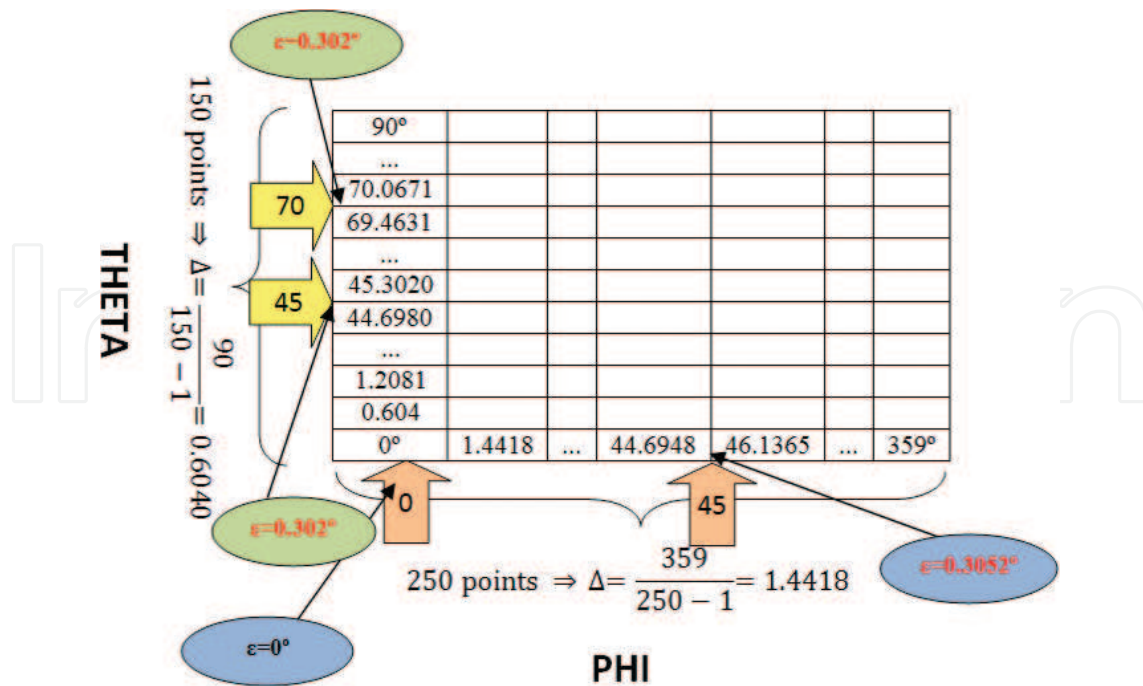
Figure 10. Evaluation grid of 2D MUSIC algorithm.

The 2D ESPRIT errors are mainly due to the mathematical process and the noise of the signal; however, in the 2D MUSIC, the errors are strongly affected by the evaluation interval, as explained in **Figure 10**.

The accuracy of the 2D MUSIC algorithm depends on the number of points of its angle grid in which Eq. (10) is evaluated. More points lead to longer computational times. The number of points of this evaluation grid is then the main issue of MUSIC algorithm, and this value should be an agreement between the required accuracy and computational load allowed.

Figure 10 shows how the MUSIC algorithm of error is consistent with the error estimates of **Figure 9**, and its relationship with the number of grid points, that in these simulations were 150 in θ and 250 in ϕ.

4. Beamforming algorithms

An antenna array, depending on the environment where it is inserted, is usually affected by several electromagnetic signals propagating around, where some are desired to be received, and others, called interferences, are undesirable. Signals of interest and intrusive signals occupy the same frequency range; however, they come from different spatial locations.

In order to improve system performance, the signals of interest must be received in the perfect way, while the impact of the unwanted interferences must be reduced or abolished.

In an adaptive antenna array, after the locations of arriving signals are identified, it is essential to use spatial-filtering techniques, also known as beamforming techniques. These techniques
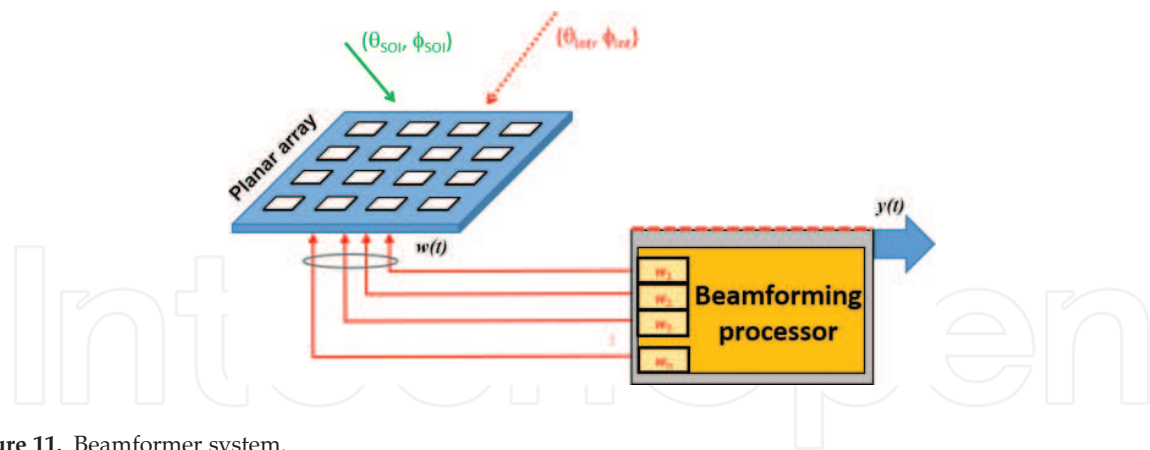
Figure 11. Beamformer system.

deal with the radiation pattern (or beam pattern) of an antenna array, shaping it according to a group of constraints, improving the performance of the communication system.

The beamforming process consists of, as is illustrated in **Figure 11**, changing the relative amplitude and phase of feeding of each element of the array, estimated through beamforming algorithms, adjust the radiation pattern of the array, steering its main lobe and putting nulls in interference directions. There are a number of developed algorithms to estimate the complex weights (amplitude and phase) to apply to an antenna array.

These algorithms are classified as data independent or statistically optimum, according to how the weights are estimated [22, 23]. In the data-independent beamforming algorithms, the weights are estimated to provide a desired response independently of the data received through the antenna, while in the statistically optimum, the weights are estimated according to the statistics of the received data, to optimize its response.

Sometimes, the statistical information of the received data is not available or changes in time; therefore, adaptive algorithms are usually applied to estimate the weights. In this situation, the calculated weights tend to a solution statistically optimum.

4.1. Data-independent algorithms

The data-independent beamformers are characterized by the estimation of its weights to be independent of the received data statistics from the array. One of the most used data-independent algorithms is the classical beamformer [22].

4.1.1. Classical beamformer

The operating principle of this method is similar to the phased array, since it estimates the necessary weights w in order that the maximum of radiation is steered to a desired direction θ_{SOI}. The vector w adjusts the phases to feed each element, in a way that the signals of all the elements are added constructively in a certain direction. However, the main limitation is the impossibility of placing nulls, which at many times is required to eliminate the impact of unwanted signals.

4.2. Statistically optimum algorithms

The statistically optimum beamformers are characterized by the calculation of its weights based on data statistics of received signals. The weights are estimated to steer the radiation pattern in the direction of the SOI while reducing the influence of interfering signals and noises. Some examples of statistically optimum algorithms mentioned in the literature [22] are the multiple side-lobe canceller (MSC), reference signal, SNR maximization and the linearly constrained minimum variance (LCMV).

4.2.1. Multiple side-lobe canceller

The MSC beamformer [22] consists of a main channel and other auxiliary channels. The main idea of this algorithm is to estimate the appropriate weights to apply to the auxiliary channels, in order to cancel or reduce the impact of interference signals in the main channel. In the main channel, a data-independent beamformer can be used to steer the maximum of the array to the desired direction.

Although this method is simple and effective, when the signal of interest is weak relating to interferences, it presents some limitations since the weights are estimated with the absence of the desired signal.

4.2.2. Reference signal

The use of reference signal beamformer [22] requires some knowledge about the desired signal to create a reference signal, and its objective is to minimize the mean square error between the output of the beamformer and the reference signal. The main difficulty of this method is to generate a proper reference signal.

4.2.3. Maximum SNR

The maximum SNR [22] method requires the knowledge of the covariance matrix of the desired signal and of the noise, and the weights are estimated to maximize the SNR.

4.2.4. Linear constrained minimum variance

One of the most important statistically optimum beamformers with higher applicability is the linear constrained minimum variance (LCMV) method [22], and a different approach of its formulation known as generalized side-lobe canceller (GSC).

Most of the times, the desired reference signal is unknown or we do not have enough information about it, being necessary to impose some linear constraints in the weight vector to minimize the variance of beamformer output. The LCMV technique can overcome the main drawbacks of most of the techniques presented before [24].

The principal idea of the LCMV beamformer is that its output is constrained in a way that signals of interest are preserved and the undesired (noise and interfering signals) are minimized.

The LCMV formulation problem is to select the complex weights that are suitable to the multiple linearly independent constraints:

$$\min_w w^H R_x w \text{ Subject to } C^H w = f \tag{11}$$

where w is the vector of weights, R_x the covariance matrix, C is the constraint matrix and f is the response vector.

The solution of the constrained minimization of LCMV problem can be obtained using the method of Lagrange multipliers [22] resulting in

$$w_{\text{opt}} = R_x^{-1} C (C^H R_x C)^{-1} f \tag{12}$$

It is important to observe the dependence of the optimal weight vector of Eq. (12) with the data correlation matrix, and therefore with the statistics of the input signal.

4.2.5. Generalized side-lobe canceller

The generalized side-lobe canceller (GSC) is a different approach to solve the LCMV problem, providing a simple implementation of the beamformer, changing the constrained minimization problem to an unconstrained arrangement [5, 25]. The GSC method, illustrated in **Figure 12**, splits the LCMV problem into two parts, one data independent and other data dependent.

This structure splits the optimum weight vector into two orthogonal components, which are in the range and in the null space of C, in such way that $w = w_0 - B\, w_M$. The beamformer output is $y = w_0^H x - w_M^H B^H x$.

The vector w_0 is designed to comply with the imposed constraints, consisting in the quiescent part of w. This weight vector is also independent of the input data and represents the non-adaptive component of the LCMV solution. Then, w_0 must satisfy the linear constraints [5]:

$$C^H w_0 = f \Rightarrow w_0 = C(C^H C)^{-1} f \tag{13}$$

The bottom branch of GSC is the data-dependent part, and consists of the blocking matrix B and w_M that will block influence of interfering signals, while minimizing the variance of the output signal y. The blocking matrix B must be orthogonal to the constraint matrix C, so $C^H B = 0$.
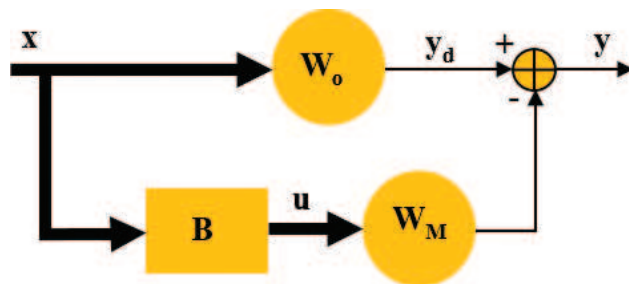

Figure 12. Generalized side-lobe canceller.

The GSC unconstrained problem is

$$\min_{w_M}(w_0 - Bw_M)^H R_x(w_0 - Bw_M) \text{ subject to } C^H w = f \tag{14}$$

and the optimal solution is

$$w_M = (B^H R_x B)^{-1} B^H R_x w_0 \tag{15}$$

This implementation of GSC beamformer enables important benefits, since the w_0 is data-independent beamformer and w_M is an unconstrained beamformer.

4.3. Adaptive algorithms

Adaptive algorithms [22] solve the problem of the statistics of the received data, which changes over the time or may not be available, and affect the statistically optimum beamformers. Examples of adaptive algorithms are the least mean square (LMS), the recursive least squares (RLS) or Frost's algorithm.

4.3.1. Frost's algorithm for LCMV beamforming

Frost's algorithm [23] belongs to the group of LCMV beamformers. Its weights are estimated based on the statistics of the received data, which sometimes are not available or are continually changing, the use of adaptive algorithms being necessary. Frost's algorithm minimizes the mean square error while maintains the specified response to the desired signal. The weight vector starts with an initial value:

$$w = C(C^H C)^{-1} f \tag{16}$$

and at each iteration, the vector is updated on negative-gradient direction with a factor μ, and the weights are defined by

$$w(n+1) = C(C^H C)^{-1} f + P(w(n) - \mu e^*(n)x(n)) \tag{17}$$

$$P = I - C(C^H C)^{-1} C^H \tag{18}$$

4.3.2. Least mean square

Least-mean-square adaptive algorithm estimates the gradient vector from the available data [22, 23, 26, 27]. This method follows an iterative procedure that successively adjusts the weight vector in the direction of the negative of the gradient vector at each iteration, eventually leading to the minimum mean square error:

$$w_M(n) = w_M(n-1) + \mu\, u(n-1)\, y^*(n-1) \tag{19}$$

$$y(n) = y_d(n) - w_M^H(n)u(n) \tag{20}$$

$$0 \leq \mu \leq \frac{1}{\lambda_{\max}}$$

where $\lambda_{\max}$ is the largest eigenvalue of the correlation matrix.

The gain μ ($0<\mu<1$) is the parameter that controls the convergence rate of the algorithm. Smaller values lead to slow convergence, but good approximation, while higher values result in a faster convergence and the stability around the minimum value is not ensured. This algorithm is simple since it requires no matrix inversion and by correctly choosing the μ-value, the weight vector tends to an optimum solution.

4.3.3. Recursive least squares

The recursive least squares algorithm has a faster convergence rate than the LMS; however, it involves more complex mathematical operations and requires more computational resources [22, 23].

The RLS problem is

$$\min_{w_{M(k)}} \sum_{n=0}^{N} \lambda^{N-n} |y_d(n) - w_M^H(n)\, u(n)|^2 \tag{21}$$

The RLS calculates the error signal $\alpha(n)$ and is given by

$$\alpha(n) = y_d(n) - w_M^H(n-1)u(n) \tag{22}$$

and then updates the weight vector according to a gain vector $k(n)$

$$w_M(n) = w_M(n-1) + k(n)\alpha^*(n) \tag{23}$$

$$k(n) = \frac{\lambda^{-1}v(n)}{1 + \lambda^{-1}u^H(n)v(n)} \tag{24}$$

The constant λ, $0<\lambda<1$ is called forgetting factor.

$$v(n) = P(n-1)u(n) \tag{25}$$

The matrix $P(n)$ corresponds to the inverse correlation matrix of the input signal which has $P(0)$ as initial value.

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)v^H(n) \tag{26}$$

$$P(0) = \delta^{-1}I \tag{27}$$

4.4. Test and comparison of the beamforming algorithms

The beamforming weights can be exhibited in the exponential form $w = Ae^{\phi}$, and consist of a set of amplitude and phases, that are applied to each element of an antenna array, to combine its signals in such a way that it produces constructive interference for the desired locals and destructive for undesired.

With the MATLAB, some beamforming algorithms were implemented, and the weights from each algorithm were computed. Then, their performance was analysed when applied to the aforementioned planar antenna array, as presented in **Figure 5**.

The system (DOA and beamforming) simulation was tested using several groups of angles of arrival with excellent results, first by estimating the necessary weights, after, applying it to the simulated planar array to analyse its radiation pattern. In the simulations presented here, it was considered that the signals were coming from signal of interest: $(\theta,\phi)=(45°, 45°)$ and interference signal: $(\theta,\phi)=(70°, 0°)$.

The LVCM algorithm was implemented, using as input parameters the two pairs of angles considered, and with a response vector $f = [1\ 0]^H$ in order to assume the first pair of angles the SOI direction, and the second the SNOI. The output weights, given in terms of amplitude and phase, to apply to the corresponding element of the 4×4 array, are presented in **Table 1(a)**. The resultant radiation pattern when these weights are applied is shown in **Figure 13(a)**.

Amplitude ∠ Phase				
	1	**2**	**3**	**4**
(a)	Optimum LCMV			
1	1.0 ∠ 0°	1.0 ∠ −90°	1.0 ∠ −180°	1.0 ∠ 90°
2	1.0 ∠ −90°	1.0 ∠ 180°	1.0 ∠ 90°	1.0 ∠ 0°
3	1.0 ∠ 180°	1.0 ∠ 89°	1.0 ∠ 0°	1.0 ∠ −90°
4	1.0 ∠ 90°	1.0 ∠ 0°	1.0 ∠ −91°	1.0 ∠ −180°
(b)	Adaptive Frost's LCMV			
1	1.0 ∠ 0°	1.0 ∠ −90°	1.0 ∠ −180°	1.0 ∠ 90°
2	1.0 ∠ −90°	1.0 ∠ −180°	1.0 ∠ 90°	1.0 ∠ 0°
3	1.0 ∠ 180°	1.0 ∠ 90°	1.0 ∠ 0°	1.0 ∠ −90°
4	1.0 ∠ 90°	1.0 ∠ 0°	1.0 ∠ −90°	1.0 ∠ −180°
(c)	Adaptive LMS			
1	1.0 ∠ 0°	1.0 ∠ −83°	0.9 ∠ 156°	0.6 ∠ 83°
2	0.7 ∠ −83°	1.0 ∠ 179°	1.0 ∠ 93°	1.2 ∠ 2°
3	0.9 ∠ −178°	0.9 ∠ 61°	0.9 ∠ 23°	0.6 ∠ −78°
4	0.9 ∠ 69°	0.6 ∠ 13°	0.6 ∠ −59°	0.6 ∠ −169°
(d)	Adaptive RLS			
1	1.0 ∠ 0°	0.7 ∠ −104°	0.5 ∠ −161°	0.9 ∠ 102°
2	1.2 ∠ −92°	1.2 ∠ 175°	0.8 ∠ 107°	0.9 ∠ −34°
3	0.8 ∠ −163°	0.3 ∠ 127°	0.8 ∠ −100°	0.8 ∠ −73°
4	0.4 ∠ 101°	0.6 ∠ −4°	0.7 ∠ −114°	0.5 ∠ −128°

Table 1. Estimated weights from beamforming algorithms.

Figure 14 shows the 2D gain distribution of the array as a function of θ and ϕ coordinates, for a better analysis of the 3D radiation patterns of **Figure 13**.

Through **Figure 13(a)** together with **Figure 14(a)**, it is possible to observe the maximum of the radiation pattern pointed to the direction of the SOI (45°, 45°), while in the direction (70°, 0°) there is a low-gain value, which reduces significantly the influence of the interference signal in this direction.

The adaptive algorithms were also implemented because they are a more realistic approach, due to the environment changes.

The adaptive Frost's algorithm was tested applying the considered pair of angles. This algorithm was processed with n = 100 samples of the input signal, iteratively. The output resulting weights are shown in **Table 1(b)**, and the produced radiation pattern is presented
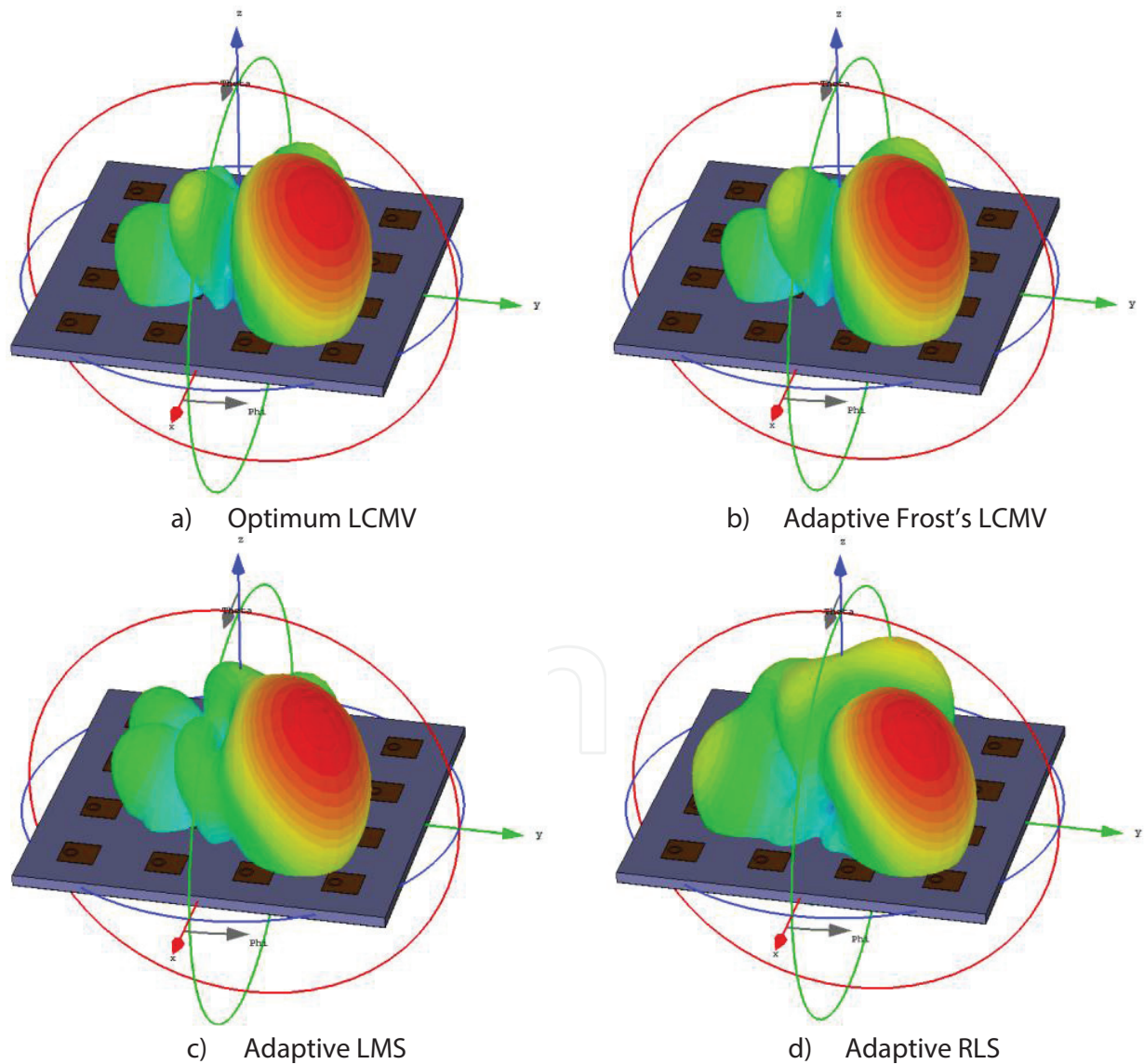

a)   Optimum LCMV   b)   Adaptive Frost's LCMV

c)   Adaptive LMS   d)   Adaptive RLS

Figure 13. 3D radiation pattern from the planar array with weights applied.

in **Figures 13(b)** and **14(b)**. The array points to the direction of interest and places a null in the interference zone, as desired.

The least-mean-square algorithm was also performed in MATLAB using 100 samples of the received signal *X(t)*. The output of this algorithm is shown in **Table 1(c)**, and the respective radiation pattern is presented in **Figure 13(c)**. Through **Figures 13(c)** and **14(c)**, it is possible to find out that using these weights, the antenna steers its main lobe in the direction of interest (45°, 45°) and places a null in the SNOI zone (70°, 0°).

Finally, the RLS algorithm was implemented and its weights for identical scenario were estimated. The output weights are shown in **Table 1(d)**, leading to the radiation pattern of **Figure 13(d)**. The array directs its maximum of radiation pattern to SOI direction (45°, 45°) as can be seen in **Figure 14(d)**.

The four beamforming algorithms being implemented and tested, a set of 50 consecutive experiments were done, performing a comparative statistical analysis, evaluating the runtime for each iteration. **Figure 15** shows the progress of the execution time of each beamforming algorithm along the *n* experiences. In each sub-figure, a straight line (red) corresponding to the average runtime of all the 50 samples is also presented.

According to **Figure 15**, there are a couple of samples with a more accentuated discrepancy, observed by a couple of peaks in the runtime compared to the average value, property that is common in all beamforming algorithms.
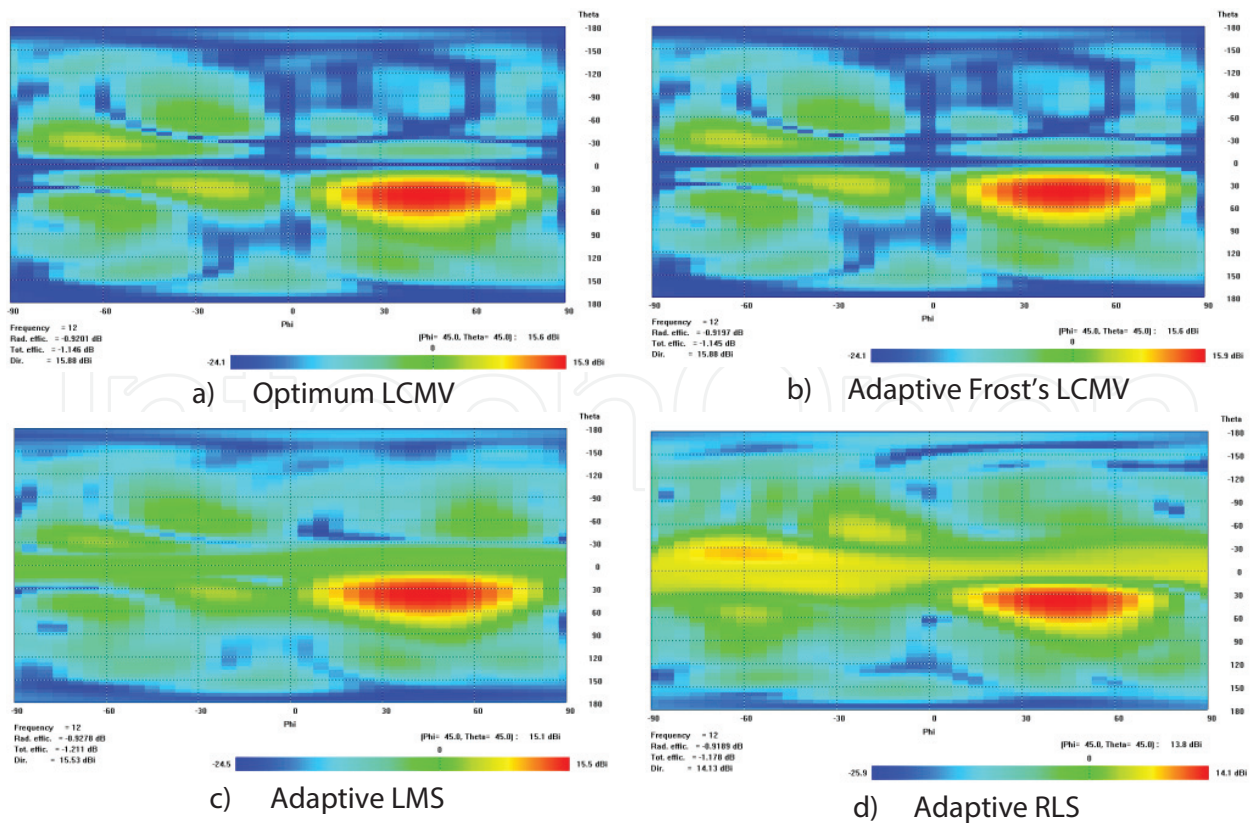

Figure 14. 2D gain distribution of the radiation pattern from the planar array with weights applied.
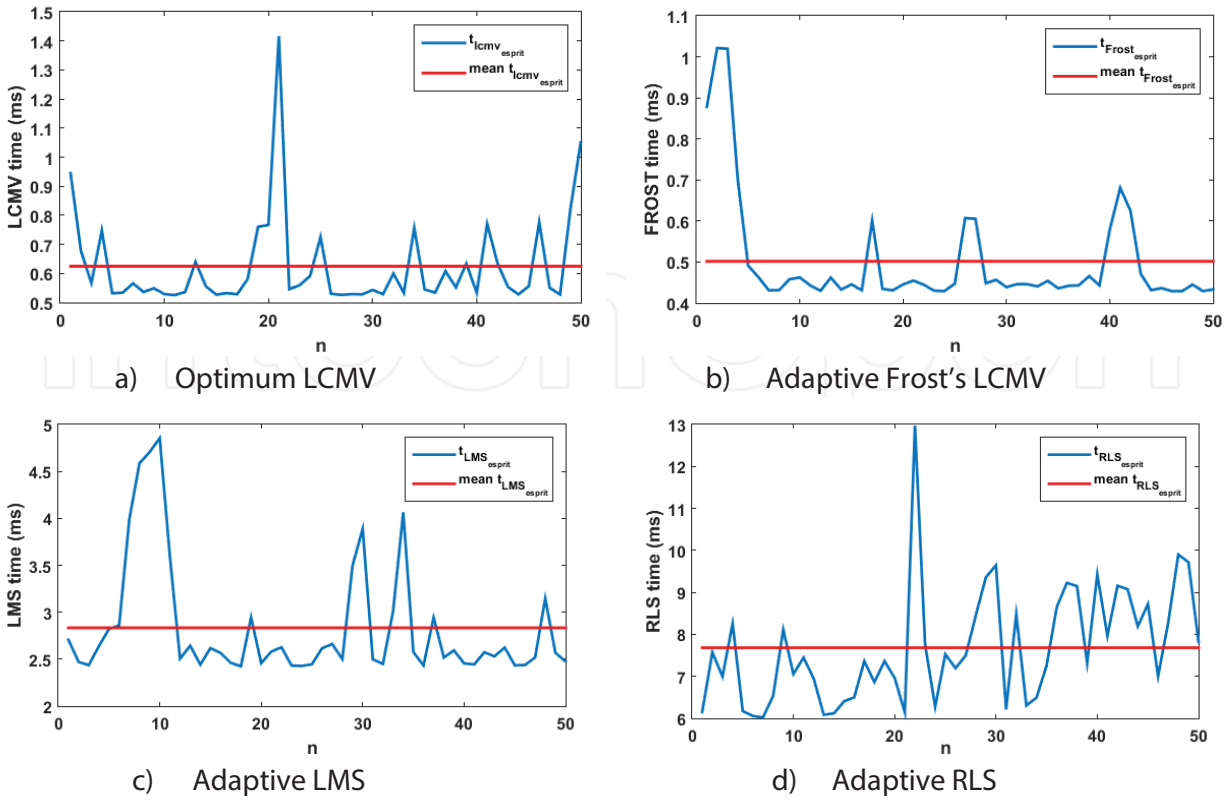
Figure 15. 3D radiation pattern from the planar array with weights applied.

The LCMV algorithm, **Figure 15(a)**, has an average runtime of about 0.62 ms. Concerning the adaptive algorithms, Frost's shown in **Figure 15(b)** present a mean value of 0.5 ms, that even being an adaptive presents a better result than the LCMV (but similar).

The LMS and RLS algorithms, presented in **Figure 15(c)** and, have an average runtime over all the 50 experiences, of about 2.8 and 7.6 ms, with the RLS, as expected, requiring more computational resources than LMS.

After the evaluation of the DOA and beamforming algorithms independently, a comparative analysis about their performance in the system was made, operating together as an adaptive antenna array. The system consists of DOA estimation followed by beamforming weights computation. It was evaluated in terms of total runtime (for all algorithms), the changeability of its results when the noise level modifies, and in terms of estimation errors. The results are presented in **Tables 2–5**, for each beamforming algorithm.

According to **Table 2**, the runtime of LCMV algorithm is similar using both DOA algorithms (MUSIC and ESPRIT), and its variation with SNR is not significantly noted when SNR changes from 10 to 15 dB. However, it almost doubles when SNR changes from 10 to 5 dB. Another important and already referred characteristic is that the MUSIC algorithm is heavier than the ESPRIT in terms of runtime, and while the τ_{MUSIC} does not show a significant variation, the τ_{ESPRIT} tends to reduce with the increase of SNR. In terms of DOA estimation errors, they tend to reduce with the increase of the SNR.

Algorithms		SNR (dB)		
DOA	Performance	5	10	15
MUSIC	τ_{MUSIC} (s)	5.92	6.03	5.34
	τ_{LCMV} (s)	0.0199	0.00089	0.00098
	ε_θ	0.302	0.302	0.302
	ε_ϕ	0.305	0.305	0.305
ESPRIT	τ_{ESPRIT} (s)	0.052	0.038	0.013
	τ_{LCMV} (s)	0.0013	0.0011	0.0009
	ε_θ	0.08	0.04	0.02
	ε_ϕ	0.167	0.141	0.0275

Table 2. Variation of the runtime and estimation error, using LCMV beamforming algorithm.

Algorithms		SNR (dB)		
DOA	Performance	5	10	15
MUSIC	τ_{MUSIC} (s)	5.67	5.15	5.53
	τ_{LCMV} (s)	0.0118	0.0047	0.0007
	ε_θ	0.302	0.302	0.302
	ε_ϕ	0.305	0.305	0.305
ESPRIT	τ_{ESPRIT} (s)	0.015	0.0018	0.0077
	τ_{LCMV} (s)	0.013	0.001	0.00086
	ε_θ	0.140	0.042	0.042
	ε_ϕ	0.0188	0.014	0.010

Table 3. Variation of the runtime and estimation error, using Frost's beamforming algorithm.

Regarding ESPRIT algorithm, a reduction of the estimation errors is observed; however, using the MUSIC algorithm this error maintains constant. This fact is due to the choice of the angle grid for the evaluation of music function P_{MUSIC}, Eq. (10), as was previously referred. Its interval must be an agreement between runtime and estimation error.

Using Frost's algorithm, **Table 3**, its runtime reduces with the SNR. The performance of DOA algorithms maintains with the already described characteristics, in terms of execution time and estimation errors.

The performance using LMS algorithm is presented in **Table 4**. The runtime decreases when the SNR increases, whereas the MUSIC and ESPRIT algorithms maintain with similar characteristics to the preceding cases.

Finally, using the RLS algorithm, **Table 5**, it is possible to observe a high reduction of the runtime when SNR increases from 5 to 10 dB. The error follows the expected behaviour,

reducing when SNR increases, using the DOA ESPRIT algorithm, and with the DOA MUSIC algorithm the estimation error holds up constant.

Algorithms		SNR (dB)		
DOA	Performance	5	10	15
MUSIC	τ_{MUSIC} (s)	5.35	5.37	5.46
	τ_{LCMV} (s)	0.020	0.007	0.002
	ε_θ	0.302	0.302	0.302
	ε_ϕ	0.305	0.305	0.305
ESPRIT	τ_{ESPRIT} (s)	0.0095	0.0012	0.0011
	τ_{LCMV} (s)	0.0054	0.0050	0.0046
	ε_θ	0.203	0.042	0.02
	ε_ϕ	0.090	0.045	0.024

Table 4. Variation of the runtime and estimation error, using LMS beamforming algorithm.

Algorithms		SNR (dB)		
DOA	Performance	5	10	15
MUSIC	τ_{MUSIC} (s)	5.82	5.75	5.71
	τ_{LCMV} (s)	0.0163	0.0070	0.0161
	ε_θ	0.302	0.302	0.302
	ε_ϕ	0.305	0.305	0.305
ESPRIT	τ_{ESPRIT} (s)	0.00181	0.00180	0.0015
	τ_{LCMV} (s)	0.0101	0.01	0.01
	ε_θ	0.2294	0.0378	0.0972
	ε_ϕ	0.156	0.031	0.01

Table 5. Variation of the runtime and estimation error, using RLS beamforming algorithm.

5. Conclusions

In this chapter, a set of DOA and beamforming algorithms was studied and implemented. Furthermore, an analysis of each algorithm was carried out, allowing to understand and consolidate the differences between the various algorithms. Regarding the DOA algorithms, both have good estimates in all the simulations performed. It was possible to conclude the reason why the MUSIC algorithm has much larger runtimes and estimation errors; this is due to the grid in which the algorithm is applied. However, it was also concluded that it is a simpler algorithm to implement compared to the ESPRIT. In terms of beamforming algorithms, four algorithms have

been implemented and tested and the main notes retained are that all of them present good estimations of weights, and the radiation patterns obtained point to the desired direction. It is also possible to remark the reduction of the runtime of the algorithms with an increase of SNR.

Acknowledgements

Author details

Tiago Varum[1,2]*, João N. Matos[1,2] and Pedro Pinho[1,3]

*Address all correspondence to: tiago.varum@ua.pt

1 Instituto de Telecomunicações, Aveiro, Portugal

2 Universidade de Aveiro, Aveiro, Portugal

3 Instituto Superior de Engenharia de Lisboa, Lisboa, Portugal

References

[1] T. Varum, J. N. Matos and P. Pinho. Detect and pointing algorithm's performance for a planar smart antenna array: a review. ACES Journal. 2015;**30**(8):824–837.

[2] C. Balanis and P. Ioannides. Introduction to Smart Antennas. Synthesis Lectures on Antennas. Morgan & Claypool Publishers, USA; 2007.

[3] B. Widrow, P. E. Mantey, L. J. Griffiths and B. B. Goode. Adaptive antenna systems. Proceedings of the IEEE. 1967;**55**:2143–2159.

[4] S. J. Yu and J. H. Lee. Design of two-dimensional rectangular array beamformers with partial adaptivity. IEEE Transactions on Antennas and Propagation. 1997;**45**:157–167.

[5] J. H. Lee and Y. H. Lee. Two-dimensional adaptive array beamforming with multiple beam constraints using a generalized sidelobe canceller. IEEE Transactions on Signal Processing. 2005;**53**(9):3517–3529.

[6]   L. C. Godara. Applications of antenna arrays to mobile communications. I. Performance improvement, feasibility, and system considerations. Proceedings of the IEEE. 1997;**85**(7):1031–1060.

[7]   R. Li, C. Rao, L. Dai and S. Zhao. Adaptive-adaptive beamforming algorithm of planar array based on one-dimensional auxiliary beam. 3rd International Congress on Image and Signal Processing (CISP). 2010;**7**:3300–3303.

[8]   MATLAB. Version 7.10.0 (R2010a). Natick, MA: The MathWorks Inc.; 2010.

[9]   Ansoft Corporation. User's Guide: High Frequency Structure Simulator, Edition: REV 1.0 Software Version 10.0; 2005.

[10] A. S. Foutz and M. Banavar. Narrowband direction of arrival estimation for antenna arrays. Synthesis Lectures on Antennas: Morgan & Claypool Publishers, USA; 2008.

[11] L. C. Godara. Application of antenna arrays to mobile communications. II. Beamforming. Proceedings of the IEEE. 1997;**85**(8):1195–1245.

[12] R. Schmidt. Multiple emitter location and signal parameter estimation. IEEE Transactions on Antennas and Propagation. 1986;**34**(3):276–280.

[13] S. Sekizawa. Estimation of arrival directions using music algorithm with a planar array. ICUPC '98. IEEE 1998 International Conference on Universal Personal Communications. 1998;**1**(10):555–559.

[14] K. V. Rangarao and S. Venkatanarasimhan. Gold-music: a variation on music to accurately determine peaks of the spectrum. IEEE Transactions on Antennas and Propagation. 2013;**61**(4):2263–2268.

[15] C. C. Yeh, J. H. Lee and Y. M. Chen. Estimating two-dimensional angles of arrival in coherent source environment. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1989;**37**(1):153–155.

[16] Y. M. Chen. On spatial smoothing for two-dimensional direction-of-arrival estimation of coherent signals. IEEE Transactions on Signal Processing. 1997;**45**(7):1689–1696.

[17] Z. Chen, G. Gokeda and Y. Yu. Introduction to Direction-of-Arrival Estimation. Artech House Signal Processing Library: Artech House, USA; 2010.

[18] M. Haardt, M. D. Zoltowski, C. P. Mathews and J. Nossek. 2d Unitary esprit for efficient 2d parameter estimation. ICASSP-95, International Conference on Acoustics, Speech, and Signal Processing. 1995;**3**(3):2096–2099.

[19] M. D. Zoltowski, M. Haardt and C. P. Mathews. Closed-form 2-d angle estimation with rectangular arrays in element space or beamspace via unitary esprit. IEEE Transactions on Signal Processing. 1996;**44**(2):316–328.

[20] Y. Y. Wang and W. W. Chen. A low complexity 2d-doa estimation algorithm using signal decomposition. 4th International in High Speed Intelligent Communication Forum (HSIC). 2012;(5):1–4.

[21] J. Hui and Y. Gang. An improved algorithm of ESPRIT for signal DOA estimation. 2012 International Conference on Industrial Control and Electronics Engineering (ICICEE). 2012;(8):317–320.

[22] B. D. V. Veen and K. M. Buckley. Beamforming: a versatile approach to spatial filtering. IEEE ASSP Magazine. 1988;**5**(4):4–24.

[23] W. Liu and S. Weiss. Wideband Beamforming: Concepts and Techniques. Wireless Communications and Mobile Computing. Wiley, United Kingdom; 2010.

[24] F. Huang, W. Xing Sheng and X. Feng Ma. Efficient parallel adaptive beamforming algorithm for planar array system. 8th International Symposium on Antennas, Propagation and EM Theory. 2008;(11):282–285.

[25] A. M. G. Guerreiro, A. D. D. Neto and F. A. Lisboa. Beamforming applied to an adaptive planar array. Radio and Wireless Conference RAWCON 98. 1998:209–212.

[26] S. Razia, T. Hossain and M. A. Matin. Performance analysis of adaptive beamforming algorithm for smart antenna system. 2012 International Conference on Informatics, Electronics Vision (ICIEV). 2012;(5):946–949.

[27] S. K. Imtiaj, I. S. Misra and R. Biswas. Performance comparison of different adaptive beamforming algorithm in smart antennas. 5th International Conference on Computers and Devices for Communication (CODEC). 2012;(12):1–4.