非公開 (ATR Confidential)

TR-WEL-0002 003

Concurrent Criterion – and Algorithm – Diversity Blind Adaptive Beamforming of an Espar Antenna

Gael Sapience, Tomoyuki Aono
Keisuke Higuchi and Takashi Ohira

2004.4.2

(株)国際電気通信基礎技術研究所
波動工学研究所
〒619-0288 京都府相楽郡精華町光台二丁目 2 番地 2
Tel: 0774-95-1501 Fax: 0774-95-1508

Advanced Telecommunications Research Institute International
Wave Engineering Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
Telephone: +81-774-95-1501 Fax: +81-774-95-1508

Abstracts and Keywords

As an ESPAR (Electronically Steerable Parasitic Array Radiator) antenna has only a single-port output and parasitic elements surrounded with it, the simple configuration of it causes difficulties to apply the beamforming techniques of conventional adaptive array antennas. Several adaptive beamforming algorithms and criteria for ESPAR antennas have been proposed, though any combination of an algorithm and a criterion cannot overcome the trade-off problem between its fast convergence and its stability.

This paper presents a new blind adaptive beamforming scheme using concurrent algorithm and criterion diversity. Using algorithm diversity switching two algorithms (Steepest Gradient Algorithm and Sequential Random Algorithm), each of them using criterion diversity (M2M4 and MMMC), was found to provide faster blind beamforming with a more stable convergence.

Key words: ESPAR antenna, Blind adaptive beamforming, Steepest Gradient Algorithm, Sequential Random Algorithm, criterion diversity.

Contents

List of Figures

List of Tables

Glossary

ABF: Aerial BeamForming

BNR: Beam to Null Ratio

CDF: Cumulative Distribution Function

DBF: Digital BeamForming

DoA: Direction of Arrival

ESPAR: Electronically Steerable Parasitic Array Radiator

IEEE: Institute of Electrical and Electronics Engineers

IEICE: Institute of Electronics, Information and Communication Engineers

M2M4: 2^{nd} order Moment and 4^{th} order Moment

MCCC: Maximum Cross Correlation Coefficient

MMMC: Maximum M^{th} order Moment Criterion

QPSK: Quadrature Phase Shift Keying

SGA: Steepest Gradient Algorithm

SINR: Signal to Interference Noise Ratio

SIR: Signal to Interference Ratio

SNR: Signal to Noise Ratio

SPSA: Simultaneous Perturbation Stochastic Approximation

SRA: Sequential Random Algorithm

Notations

In this part, the notations used in this report for all equations and calculus are settled.

A scalar or complex value is noted: x, x_n, X.

A vector is noted: $\mathbf{x}$.

A matrix is noted: $\mathbf{X}$.

The identity matrix of order M is noted: $\mathbf{I}_M$.

The superscript * is the conjugate operation.

The superscript ^{T} is the transpose operation for vector or matrix.

The superscript ^{H} is the transpose and conjugate operation for vector or matrix.

The operator $|\;|$ is the modulus operation.

Introduction

Adaptive array antennas are an emerging technology that has gained much attention due to its ability to significantly increase the performance of wireless systems. However few research has been focused on the way to improve mobile terminals using advanced adaptive antennas methods. Recently, the electronically steerable parasitic array radiator (ESPAR) antenna has been proposed for compact adaptive beamforming applied to wireless communication systems [1].

The basic principle of a (M+1)-element ESPAR antenna is that only one active radiator is connected to the receiver. The M remaining elements are parasitic ones. The antenna pattern is formed according to the values of loaded reactances on these parasitic radiators. Signals on the parasitic elements cannot be observed so only the single-port output can be measured and processed as feedback to adjust the reactances. Hence, most of the algorithms made for conventional adaptive arrays cannot be directly applied to the ESPAR antenna.

Consequently, several adaptive beamforming algorithms and criteria for ESPAR antennas have been proposed. These algorithms are the Steepest Gradient Algorithm (SGA) [2], the Sequential Random Algorithm (SRA) [3], and so on. These criteria are the 2^{nd} order Moment and 4^{th} order Moment (M2M4) [4]criterion, the Maximum M^{th} order Moment Criterion (MMMC) [5] and so on. Nevertheless any combination of an algorithm and a criterion cannot overcome the trade-off problem between fast convergence and stability.

In this paper, the objective is to develop an adaptive algorithm that allows the ESPAR antenna to steer its beam and automatically in order to make the output SIR (Signal-to-Interference Ratio) as large as possible. This algorithm is based on two existing algorithms : SGA and SRA. Each of these adaptive algorithms uses criterion diversity [6], that switches between two existing blind criteria: M2M4 and MMMC.

The organization of this paper is as follows. In the first section, ATR and especially Adaptive Communications Laboratories (ACR) will be briefly described.

Then, in the second part, basic configuration and formulation of the ESPAR antenna will be explained. In Section 3, criterion diversity applied for adaptive algorithms will be developed in three parts: basic principle of criterion diversity and then SGA and SRA improved using this technique. In section 4, diversity of both algorithms and criteria is explained in two parts. First, the adaptive control principle and then simulations results are presented. Section 5 includes the concluding remarks.

Chapter 1

ATR Adaptive Communications Research Laboratories

1.1 Introduction to ATR

Advanced Telecommunications Research Institute International (ATR) was established in 1986 with support from industry, academia and government. It operates with the following objectives:

- to undertake basic, innovative research on telecommunications,
- to coordinate joint research project involving industry, academia and government,
- to implement ATR's research activities on a global basis,
- to play a guiding role in Kansai Science City based on experience gained as the first research institute.

Since its foundation, ATR conducted research activities with financial contributions from KTC, the Japan Key Technology Center, and many private companies to its research laboratories. In the middle of fiscal year 2001, the research funding system was shifted from KTC to a governmental organization called TAO, the Telecommunications Advancement Organization of Japan, which funds contracted research topics in telecommunications for a better society in the 21st century.

The reorganized structure leads to the eight following actual departments:

- Adaptive Communications Research Laboratories (ACR),
- Brain Activity Imaging Center (BAIC),
- Computational NeuroSciences Laboratories (CNS),
- Intelligent Robotics and Communication Laboratories (IRC);
- Human Information Science Laboratories (HIS),
- Media Information Science Laboratories (MIS),
- Spoken Language Translation Research Laboratories (SLT).
- Technology Liaison Center (TLC),

1.2 ACR Laboratories

ACR laboratories are engaged in research on technologies for the construction and evaluation of adaptive systems with the aim of making systems adaptive to various operating and user environments.

Focusing on wireless technologies, ACR laboratories are conducting research on fundamental technologies for user-friendly communications systems where users need not concern themselves with the complexities of how networks work. Currently, key technologies are emphasized for wireless ad hoc networks, consisting of mobile nodes which do not need the assistance of centralized infrastructure such as base stations.

1.2.1 Organisation

The organisation of ACR laboratories is depicted in figure 1.1.

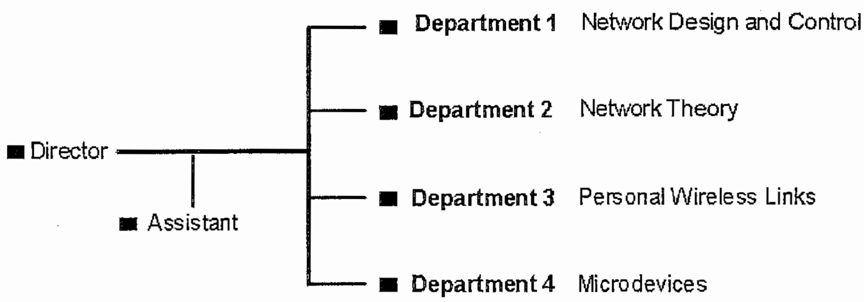

Figure 1.1: Organisation of ACR

1.2.2 Research domains

The main research domains of ACR laboratories are shown in figures 1.2, 1.3 and 1.4.

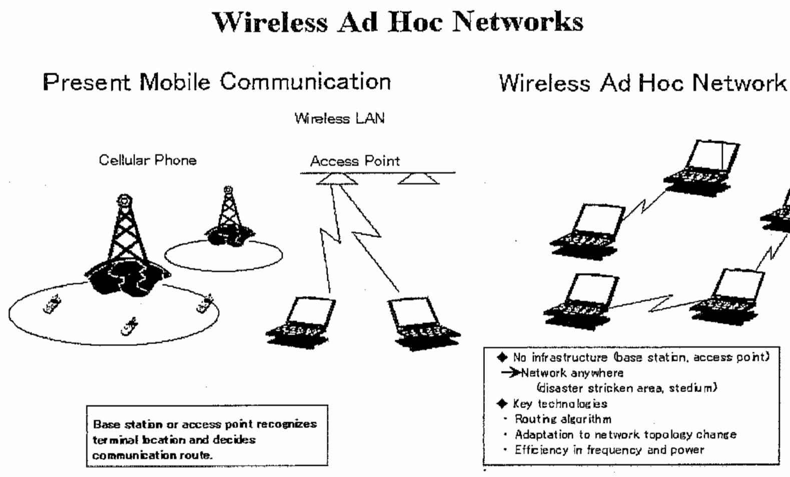

Figure 1.2: Research domains

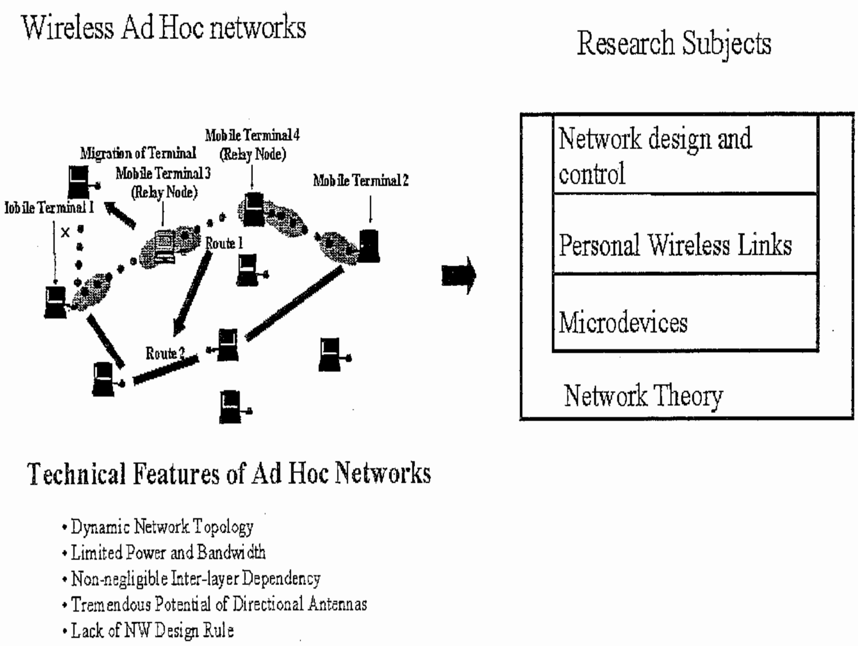

Figure 1.3: Research subjects

Expected Performance of Wireless Ad Hoc Networks

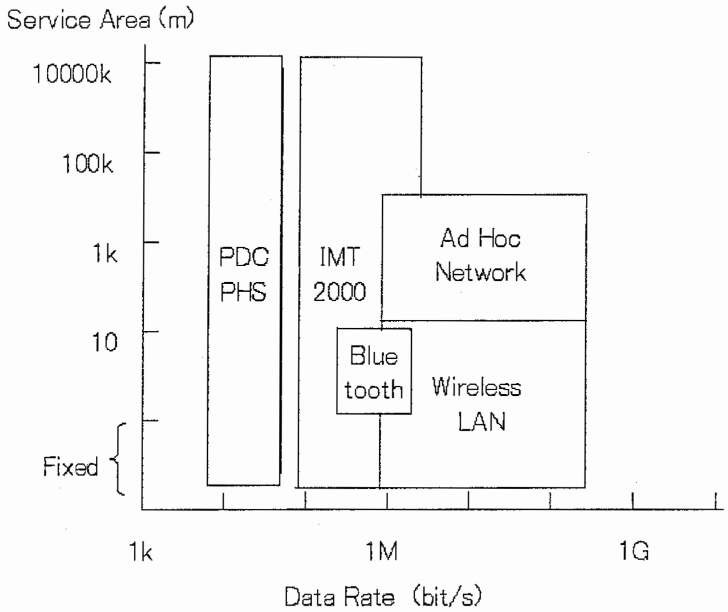

Figure 1.4: Excpected performances

In department 3, research is being carried out on smart antennas, promising key components for future wireless ad hoc networks.

In fact, an ESPAR (electronically steerable parastic array radiator) antenna is developed as the smart antenna. Since the ESPAR antenna steers autonomously its beam toward the arrival direction of desired radio waves and steers the nulls of the beam toward the undesired interfering waves, power-savings and frequency reusability in ad hoc networks can be greatly improved.

Chapter 2

Adaptive ESPAR antenna

2.1 ESPAR antenna and signal model

As a practical hardware realization of the ABF concept, the Electronically Steerable Parasitic Array Radiator (ESPAR) antenna has been proposed. In this section, ESPAR antenna configuration and signal modelisation are reviewed [2].

2.1.1 ESPAR antenna configuration

The ESPAR antenna consists of $(M+1)$ elements with $M = 6$ in our configuration. For example, a 7-element ESPAR antenna is depicted in Figure 2.1.

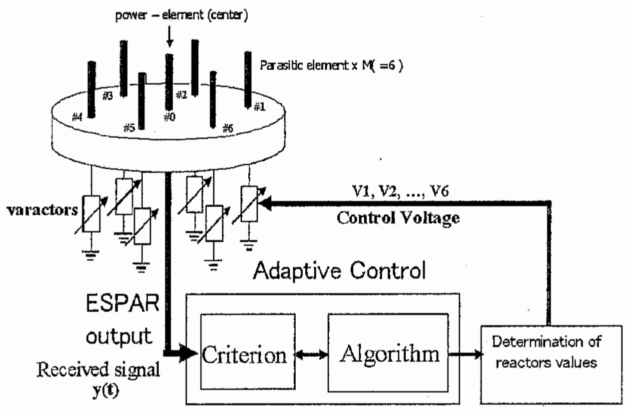

Figure 2.1: Configuration of an ESPAR antenna

This antenna is a reactively controlled array with only one active radiator. The 0-th element is the active radiator on a quarter-wavelength monopole located at the center of a circular ground plane. The M remaining parasitic elements symmetrically surrounding the active radiator. The pattern of this antenna is controlled by adjusting the values x_m, m $\epsilon(1, \cdots, M)$ of the reactances connected to the parasitic radiators.

In practical applications, the reactances may be constrained in certain ranges, *e.g.*, from $-300\ \Omega$ to $300\ \Omega$. Consequently, the vector denoted by

$$\mathbf{x} = [x_1, x_2, \cdots, x_m, \cdots, x_M]^T \tag{2.1}$$

is called the reactance vector.

Now, output formulation of the ESPAR antenna can be determined. Lets denote $\mathbf{s}(t) = [s_0(t), s_1(t), \cdots, s_m(t), \cdots, s_M(t)]^T$, where the component $s_m(t)$ is the RF signal impinging on the m^{th} element and $\mathbf{i} = [i_0, i_1, \cdots, i_M]^T$, where i_m is the RF current appearing on the m^{th} element. Therefore the single-port output $y(t)$ of the antenna is formulated in Eq. 2.2 .

$$y(t) = \mathbf{i}^T\mathbf{s}(t), \tag{2.2}$$

2.1.2 ESPAR antenna model

For convenience, the model of the ESPAR antenna is derived in a transmit mode (theorem of reciprocity tells us that all the following results are also available in the receive-mode). Now RF current vector $\mathbf{i}$ will be derived as a function of the reactance vector.

The RF voltage of the central element, can be imposed as function of the RF current appearing on this element :

$$v_0 = V_s - Z_0 i_0, \tag{2.3}$$

where Z_0 is the output impedance, and V_s is the internal source RF voltage. In the same way we impose the RF voltage on the reactance x_m to be

$$v_m = -jx_m i_m, \quad m = 1, 2, \ldots, M. \tag{2.4}$$

So we can form the RF voltage vector $\mathbf{v} = [v_0, \ldots, v_m, \ldots, v_M]^T$. Then, the RF voltage vector can be explained as a function of $\mathbf{i}$:

$$\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \\ \vdots \\ v_M \end{bmatrix} = \begin{bmatrix} V_s - Z_0 i_0 \\ -jx_1 i_1 \\ \vdots \\ -jx_m i_m \\ \vdots \\ -jx_M i_M \end{bmatrix} = V_s \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \mathbf{X}\mathbf{i}, \tag{2.5}$$

where $\mathbf{X} = diag[Z_0, jx_1, \ldots, jx_M]$ is called the reactance matrix. In addition, the RF current vector $\mathbf{i}$ and RF current voltage $\mathbf{v}$ have the relationship :

$$\mathbf{i} = \mathbf{Y}\mathbf{v}, \tag{2.6}$$

where $\mathbf{Y}$ is the admittance matrix, with

$$\mathbf{Y} = \begin{bmatrix} y_{00} & \cdots & y_{0M} \\ & \ddots & \\ \vdots & y_{kl} & \vdots \\ & \ddots & \\ y_{M0} & \cdots & y_{MM} \end{bmatrix} \tag{2.7}$$

and y_{kl} represents the mutual admittance between the elements k and l for $(k,l) \in [0,M]^2$.

By replacing the voltage vector in (2.6) by its expression in (2.5), the current vector becomes :

$$\mathbf{i} = V_s(\mathbf{I}_{M+1} + \mathbf{YX})^{-1}\mathbf{y}_0 \tag{2.8}$$

where $\mathbf{I}_{M+1}$ is the identity matrix of order $M+1$. Moreover, the $(M+1)$-dimensional vector $\mathbf{y}_0$ and $(M+1)$ by $(M+1)$ matrix $\mathbf{Y}$ are constant and determined by the structure of the ESPAR antenna. Notice that this current vector $\mathbf{i}$, and thus $y(t)$ is a function of the reactance vector $\mathbf{x}$ of Eq. (2.1.1).

By the reciprocity theorem, it holds, similarly to conventional array antennas, that

$$\forall (k,l) \in [0,M]^2, \quad y_{kl} = y_{lk}. \tag{2.9}$$

In addition, the cyclic symmetry of the elements of the ESPAR antenna implies

$$\begin{gathered} y_{11} = y_{22} = y_{33} = y_{44} = y_{55} = y_{66} \\ y_{01} = y_{02} = y_{03} = y_{04} = y_{05} = y_{06} \\ y_{12} = y_{23} = y_{34} = y_{45} = y_{56} = y_{61} \\ y_{13} = y_{24} = y_{35} = y_{46} = y_{51} = y_{62} \\ y_{14} = y_{25} = y_{36} \end{gathered}$$

Finally, equation (2.9) implies that the admittance matrix $\mathbf{Y}$ is determined by only 6 components of the mutual admittances [2] y_{00}, y_{10}, y_{11}, y_{21}, y_{31} and y_{41}. The values of these 6 components depend on the physical structure of the antenna, *e.g.*, the radius, the space intervals and the lengths of the elements, and therefore are constant.

Summarizing the above explanation, we write the admittance matrix $\mathbf{Y}$ as

$$\mathbf{Y} = \begin{bmatrix} y_{00} & y_{10} & \cdots & \cdots & \cdots & \cdots & y_{10} \\ y_{10} & y_{11} & y_{21} & y_{31} & y_{41} & y_{31} & y_{21} \\ \vdots & y_{21} & \ddots & \ddots & \ddots & \ddots & y_{31} \\ \vdots & y_{31} & \ddots & \ddots & \ddots & \ddots & y_{41} \\ \vdots & y_{41} & \ddots & \ddots & \ddots & \ddots & y_{31} \\ \vdots & y_{31} & \ddots & \ddots & \ddots & \ddots & y_{21} \\ y_{10} & y_{21} & y_{31} & y_{41} & y_{31} & y_{21} & y_{11} \end{bmatrix}. \tag{2.10}$$

Similarly, the $\mathbf{y}_0$ vector can be rewritten as $\mathbf{y}_0 = [y_{00}, y_{10}, y_{10}, \ldots, y_{10}]^T$.

The $\mathbf{i}$ current vector can also be formulated as [2]

$$\mathbf{i} = V_s(\mathbf{Z} + \mathbf{X})^{-1}\mathbf{u}_0, \tag{2.11}$$

where the $(M+1)$-dimensional vector $\mathbf{u}_0$ is equal to $[1, 0, \ldots, 0]^T$. Moreover, the $(M+1)$ by $(M+1)$ impedance matrix $\mathbf{Z}$ is simply the inverse of the $\mathbf{Y}$ admittance matrix.

However, contrary to conventional adaptive array antennas, the signal vector $\mathbf{s}(t)$ impinging on the elements of the ESPAR antenna is not measurable.

For the ESPAR antenna, only the single-port output $y(t)$ can be measured. Unfortunately, this is a highly nonlinear function of $\mathbf{x}$, and includes an intractable matrix inverse, which makes it difficult to produce an analytical expression of adaptive performance. It is also interesting to note that the current vector $\mathbf{i}$ is equivalent to the weight coefficient vector of the conventional adaptive array processing. It is clear from Eq. (2.8) that each component of $\mathbf{i}$, unlike the weight vector of the conventional adaptive array processing, is not independent but mutually coupled with each other. The discussion above implies that a direct application of most of the algorithms of the conventional adaptive array to the ESPAR antenna is impractical.

2.1.3 Signal model

First of all, we give a steering vector of the ESPAR antenna. Consider the $M+1$-element ESPAR antenna as shown in Figure 2.2.

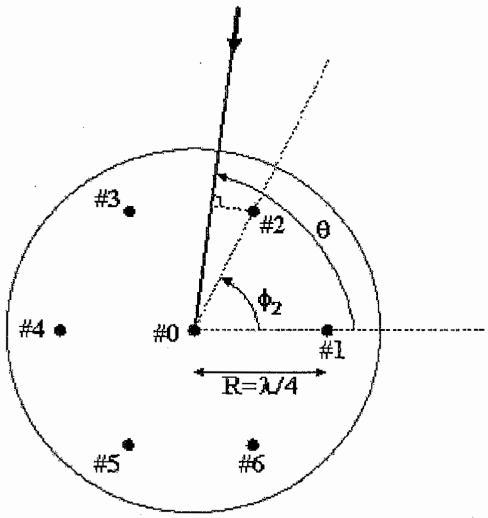

Figure 2.2: Geometry of an ESPAR antenna

The m^{th} element is placed at an angle $\phi_m = \frac{2\pi}{M}(m-1), \quad m\epsilon[1,2,\cdots,M]$ relative to an arbitrary axis. When an incoming wavefront is impinging on the antenna from a Direction-of-Arrival (DoA) of θ relative to the same reference axis, there is a spatial delay of $Rcos(\theta - \phi_m)$ between the signals received at the pair of the m-th element and the 0-th element. Considering the incident signal has a wavelength of λ, this spatial delay can be expressed as an electrical angular difference defined by $\frac{2\pi}{\lambda}Rcos(\theta - \phi_m)$.

For the ESPAR antenna the radius is $R = \frac{\lambda}{4}$ then the steering vector for a DoA of θ can be expressed as 2.12

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_1)} \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_2)} \\ \vdots \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_M)} \end{bmatrix}. \tag{2.12}$$

Suppose there are a total number of D transmitted signals $u_d(t)$ with direction-of-arrivals (DoAs) θ_d $(d = 1, \ldots, D)$. Let $s_m(t)$ $(m = 0, 1, \ldots, M)$ denote the RF signal impinging on the m^{th} element of the antenna, and let $\mathbf{s}(t)$ be the column vector with m^{th} component $s_m(t)$. Then, the column vector $\mathbf{s}(t)$ may be expressed as

$$\mathbf{s}(t) = \sum_{d=1}^{D} \mathbf{a}(\theta_d) u_d(t); \tag{2.13}$$

where $\mathbf{a}(\theta_d)$ is the steering vector depending on the structure of the ESPAR antenna for the DoA of θ_d.

According to this signal model, we can give the output of the ESPAR antenna as [2]

$$y(t) = \mathbf{i}^T \mathbf{s}(t) + n(t) = \sum_{d=1}^{D} \mathbf{i}^T \mathbf{a}(\theta_d) u_d(t) + n(t), \tag{2.14}$$

where $n(t)$ is an additive white Gaussian noise and $\mathbf{i}$ is the RF current vector.

The RF currents appearing on the elements are not independent but mutually coupled with each other, and depend on the value of the reactances. The RF current vector is $\mathbf{i}(t) = [i_0(t), \ldots, i_m(t), \ldots, i_M(t)]^T$, with the component $i_m(t)$ appearing on the m-th element. Notice that the t from the current vector $\mathbf{i}(t)$ is voluntarily omitted for the sake of simplicity and this vector will be noted $\mathbf{i}$.

Compared with conventional array antennas, ESPAR antenna has low hardware complexity, low cost and low power consumption. The ESPAR antenna, does not need RF-amplifiers, bandpass filters, A/D converters for each element like in conventional adaptive array. This specific antenna seems to be suitable for mobile applications and wireless computer network.

However, because of the configuration of the ESPAR antenna, three difficulties are encountered with the development of adaptive algorithms :

- Signals on the surrounding passive elements $[1, \cdots, M]$ cannot be observed. Only the single-port output can be measured and processed as feedback to adjust the reactances.
- The RF currents are not independent but mutually coupled with each other, and depend on the value of the reactances.
- The single-port output is a highly nonlinear functions of these variable reactances and includes an intractable matrix inverse (see Eq. 2.12).

Chapter 3

Criterion diversity applied for adaptive algorithms

3.1 Criterion Diversity

Usually, in adaptive algorithms, only one criterion is used for beamforming. Table 3.1 shows two criteria values and their characteristics for blind beamforming for ESPAR antennas. These criteria are blind, so they do not use training symbols but only the received signal. The used blind criteria are:

- the 2^{nd} order Moment and 4^{th} order Moment (M2M4).
- the Maximum M^{th} order Moment Criterion(MMMC).

In table 3.1, ρ_n represents the criterion value and N_s denotes the number of sample statistic moments. In fact, criterion ρ_n is computed from N_s samples of the array of the output signal y_n.

Furthermore, an other parameter for the MMMC is M which represents the M^{th} order of the criterion.

Unfortunately, Table 3.1 shows that all criteria have a trade-off problem between their convergence speed and their stability.

Table 3.1: Comparison of the Blind beamforming criteria for ESPAR antennas

Criterion	M2M4	MMMC
Value	$\rho_n = \frac{\sum_{n=1}^{N_s} \lvert y_n\rvert^4}{\left\{\sum_{n=1}^{N_s} \lvert y_n\rvert^2\right\}^2}$	$\rho_n = \frac{\lvert\sum_{n=1}^{N_s} y_n^M\rvert^2}{\sum_{n=1}^{N_s} \lvert y_n^M\rvert^2}$
Speed	High	Low
Stability	Low	High
SNR estimation	possible	possible

Characteristics of these blind criteria are illustrated in figure 3.1. In this one, an adaptive algorithm uses only one criterion at a time. In fact, it appears that convergence of M2M4 criterion is faster than MMMC whereas stability is better for this last one.

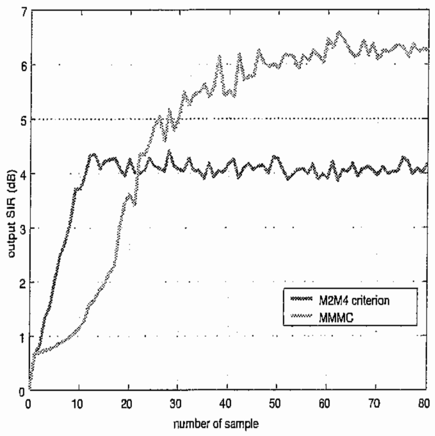

Figure 3.1: SGA convergence curves using different criteria, SNR= 10dB, DoA [0° 45°].

Consequently, to overcome this problem and to obtain a blind adaptive beamforming with faster and more stable convergence, both of these criteria can be used within adaptive algorithms: this is the basic principle of *criterion diversity*.

Usually, only one criterion is used during the blind beamforming whereas this new method determines the optimum criterion among the M2M4 and MMMC at each update of the reactances values [6].

3.2 Steepest Gradient Algorithm with criterion diversity

3.2.1 Algorithm principle

We can now describe a gradient-based algorithm with criterion diversity for the ESPAR antenna. This algorithm uses both blind criteria M2M4 and MMMC for criterion diversity (they are designated as $\rho_{n_{M2M4}}$ and $\rho_{n_{MMMC}}$). This principle is used as cost function for the steepest gradient algorithm.

First, let's define a gradient vector of $\bigtriangledown \rho_n$ which is expressed as :

$$\bigtriangledown \rho_n = \begin{bmatrix} \frac{\delta \rho_n}{\delta x_1} \\ \frac{\delta \rho_n}{\delta x_2} \\ \vdots \\ \frac{\delta \rho_n}{\delta x_M} \end{bmatrix}, \tag{3.1}$$

where $\frac{\delta \rho_n}{\delta x}$ denotes the derivative with respect to reactance vector $\mathbf{x}$.

Then, basically the steepest gradient algorithm with criterion diversity proceeds as follows:

- Step 1: Set initial values for the reactance vector $\mathbf{x}$. Generally $\mathbf{x}$ is set to a zero vector which leads to an omni-directional antenna.
- Step 2: Using the present reactance vector guess, calculate the gradient vector $\bigtriangledown \rho_n$ on two different ways using M2M4 and MMMC.
- Step 3: Compute next guess of the reactance vector, by choosing the criterion with the greatest norm on the gradient vector $\bigtriangledown \rho_n$.
- Step 4: Go back to step 2 and repeat the process during N iterations.

According to the flowchart of steepest gradient algorithm, we can observe that successive corrections to the reactance vector $\mathbf{x}$ in the directions of the positive of the gradient vector should lead to a good reactance vector.

Then, according to the steepest gradient theory, if we denote $\mathbf{x}(n)$ the value of reactance vector $\mathbf{x}$ at time n, this algorithm will be ruled by the recursive relation :

$$\mathbf{x}(n+1) = \mathbf{x}(n) + \mu \bigtriangledown \rho_n, \tag{3.2}$$

where μ is a constant positive value that controls the convergence speed [2].

However there may be some problems to compute the gradient vector $\bigtriangledown \rho_n$. In fact, as it was stated in the Introduction section, it may be difficult to analytically represent the gradient vector as a function of $\mathbf{x}$ because of the presence of the intractable matrix inverse in the representation of $y(t)$, and the signal vector impinging on the elements on the ESPAR antenna cannot be observed.

That is why, an estimate of the gradient vector of Eq.3.1 may be derived by the use of finite-difference approximation of derivatives. Specifically, the partial derivative term $\frac{\delta\rho_n}{\delta x_m}$ can be processed as :

$$\frac{\delta\rho_n}{\delta x_m} = \frac{\rho_n(x_1, \cdots, x_m + \Delta x_m, \cdots, x_M) - \rho_n(x_1, \cdots, x_m, \cdots, x_M)}{\Delta x_m}, m = 1, \cdots, M, \tag{3.3}$$

by incrementing x_m to $x_m + \Delta x_m$,whereΔx_m is the perturbation size parameter.

As shown in the Eq.3.3, only one component of the gradient vector $\bigtriangledown\rho_n$ is calculated at a time from the output of the antenna. Consequently, all the components of reactance vector $\mathbf{x}$ are sequentially perturbed in order to get one gradient vector for each iteration of Eq.3.1.Furthermore, this process has to be realized twice because of criterion diversity. In fact, $\bigtriangledown\rho_n$ is computed using M2M4 and also MMMC. Then, the criterion which has the greatest norm on $\bigtriangledown\rho_n$ will be selected. The flowchart of this adaptive algorithm is shown in figure 3.2.

It is difficult to determine analytically the performance of the ESPAR antenna. Therefore simulations are required to validate this algorithm. Iteration number is set to 10, whereas different DoA and SNR initial values will be used during simulations.

3.2.2 Simulation results

The presence of the intractable matrix in the single-port output of the ESPAR antenna causes some difficulties to describe analytically its performances. That is why simulations with the Matlab software have been conducted.

In these one, a 7-element(M=6) ESPAR antenna is employed. This choice is based on the researches on the pattern- forming capability of $2-, \cdots, 9-$ elements ESPAR antenna in [7] and [8]. It has been proved that ESPAR antennas with few elements can not steer a beam and a deep null simultaneously. In fact, the more the number of elements increases, the more the beamforming capability becomes stronger. The 7-element ESPAR antenna can form one beam and multiple nulls in almost all directions except the regions where the directions of the beam and nulls are about 30 degrees [8](the capability of a 9-element antenna is slightly better, with elements that can steer a beam and that can provide deeper nulls are possible).

First, simulations have been conducted to observe the effect of the step-size parameter μ in Eq.3.2. For this simulation, the adaptive algorithm does not use criterion diversity (only one criterion is used to compute the gradient vector),the perturbation size parameter value, the DoA and SNR are fixed.

Figure 3.3 confirms that the rate of convergence of the ESPAR antenna is dependent on the step-size parameter μ. This simulation has been conducted whereas Δx parameter is fixed and μ parameter value will be changed.

The figure on the left represents the CDF (cumulative distribution function) of the Beam to Null Ratio (BNR) of the SGA for different step sizes values. In fact, the CDF curve indicates the probability that the BNR will reach certain values. For example, the cross on the CDF means that the probability for the BNR the SGA to reach 5dB is equal to 50 %.

These statistics for 1000 trials of 10^{th} iterations of SGA. Furthermore, the figure on the right represents the output SIR compared to the number of samples.

For 10 iterations of SGA, figure 3.3 shows that as soon as the step-size reaches the value of 1500, the CDF and output SIR curves will be almost the same.

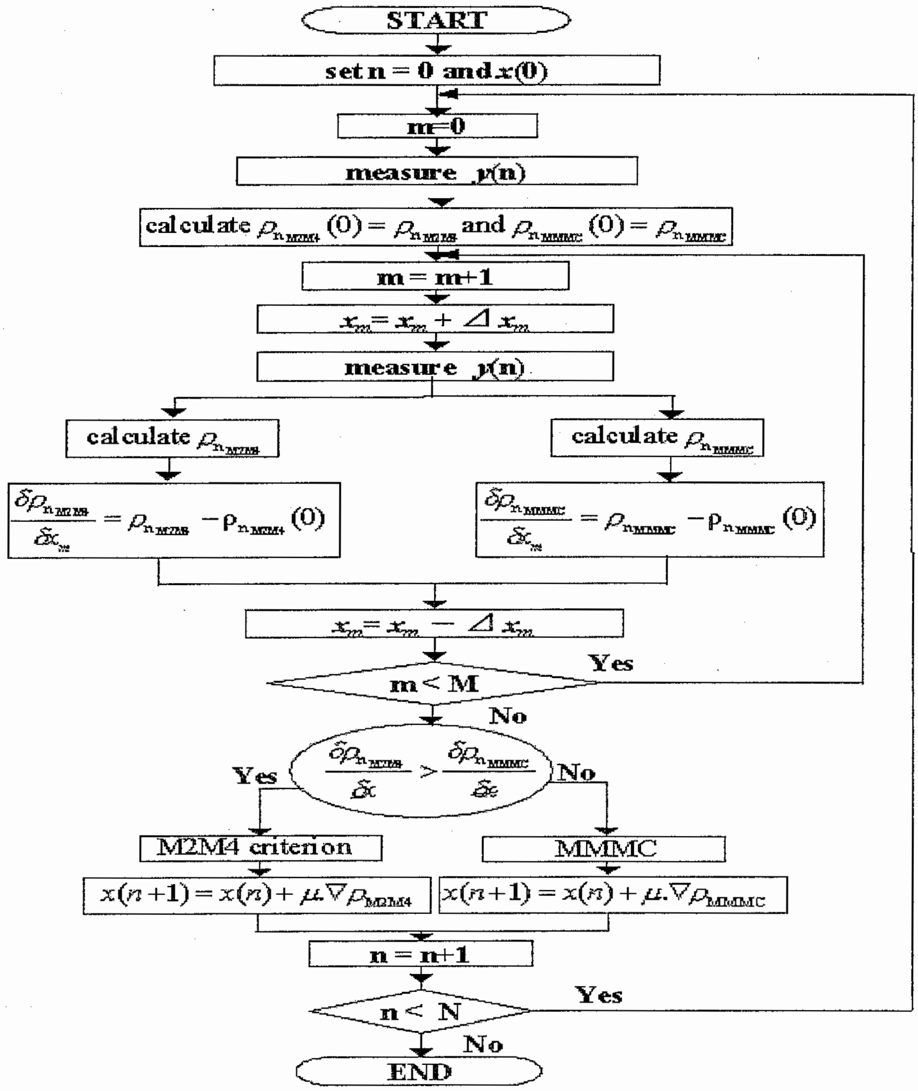

Figure 3.2: SGA with criterion diversity flowchart.

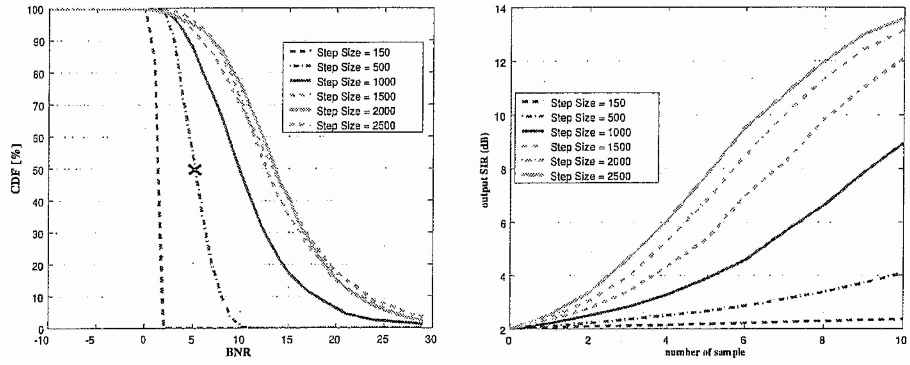

Figure 3.3: SGA CDF and convergence curves with different Step Sizes values, SNR= 10dB, DoA [0° 45°].

Then, simulations have also been conducted to observe the effect of the perturbation-size (Δx) parameter. Like the previous simulation, Δx value will change while step-size parameter will be set to 1500.

CDF and convergence curves are shown in Figure 3.4. It confirms that this parameter also affects the convergence rate. In fact due to these simulations results, Δx parameter value will be set to 32.

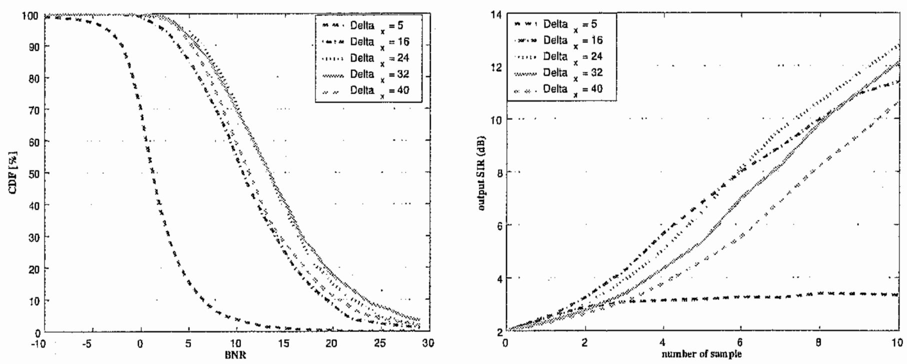

Figure 3.4: SGA CDF and convergence curves with different Delta x values, SNR= 10dB, DoA [0° 45°].

Moreover, mechanism of criterion diversity has been simulated. According to Table 1, M2M4 is the fastest criterion but not stable, contrary to MMMC. Figure 3.5 shows the criterion selection probability between M2M4 and MMMC for SGA using criterion diversity.

These statistics has been also conducted for 1000 trials of 10^{th} iterations of SGA. It shows that at the beginning of the update, M2M4 is the most frequently blind criterion selected. On the other hand, as it goes to the convergence, the MMMC is selected more often.

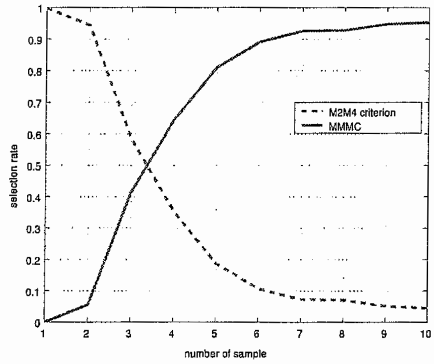

Figure 3.5: SGA selection rate curve, SNR= 10dB, DoA [0° 45°].

Figures 3.6, 3.7 and 3.8 are beam pattern results of SGA adaptive beamforming while using only M2M4 criterion, MMMC and criterion diversity.

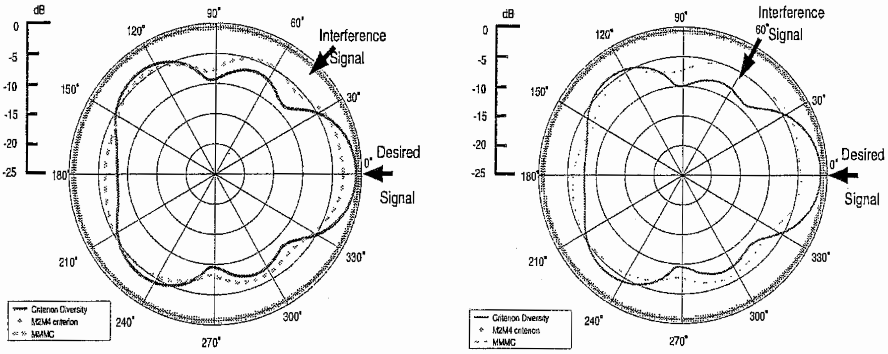

Figure 3.6: SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

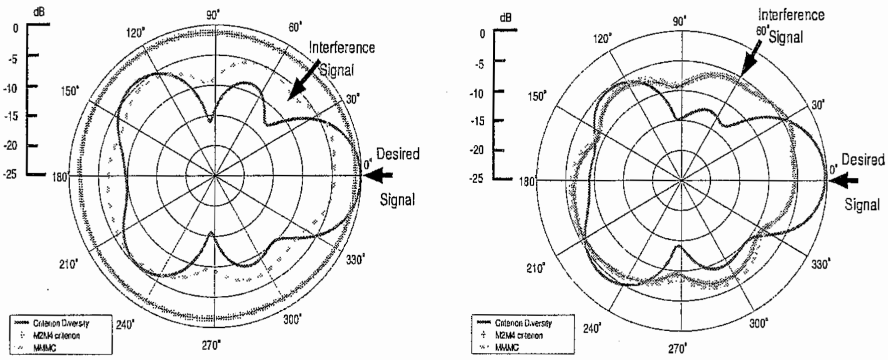

Figure 3.7: SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

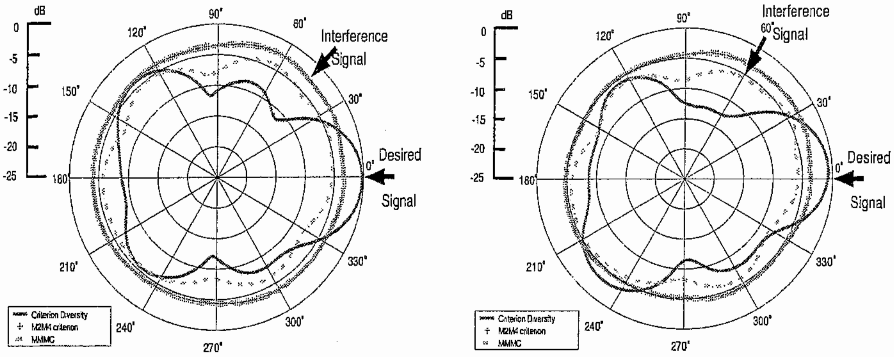

Figure 3.8: SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

In fact, simulations have been conducted for different SNR (5dB, 10dB and 20dB)and also for several DoA's: [0° 45°] and [0° 60°] (in a DoA, the first number is the direction of the signal of interest whereas the second one is direction of the interference signal).

As we can see on these patterns, the BNR is always higher for SGA using criterion diversity than for the others. That is why, it seems that criterion diversity leads to improve blind beamforming for SGA.

Figures 3.9, 3.10 and 3.11 are convergence curves in order to compare efficiency of SGA using criterion diversity against adaptive algorithms which employs only one criterion. In fact, output SIR is compared between these algorithms with average of 1000 trials of 10^{th} iterations for SGA. In fact, these figures show that whatever are the SNR and DoA's initial values, accuracy of SGA with criterion diversity is better. Consequently, criterion diversity leads to a better beamforming compared to using one criterion only.

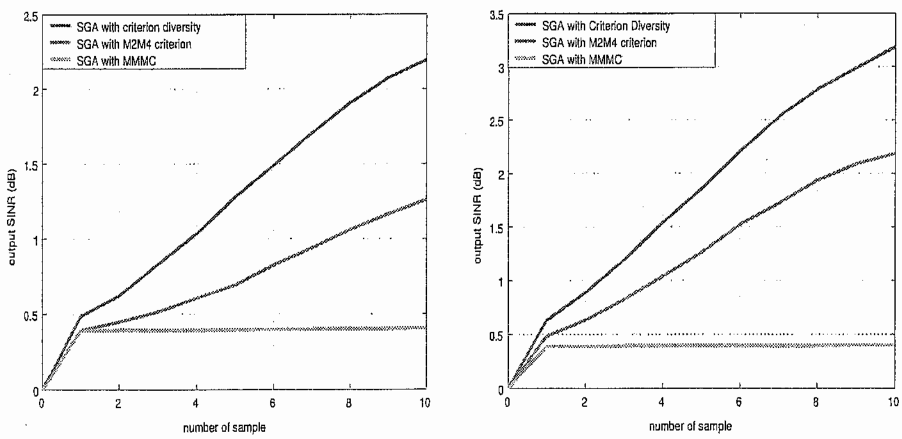

Figure 3.9: SGA convergence curves, SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

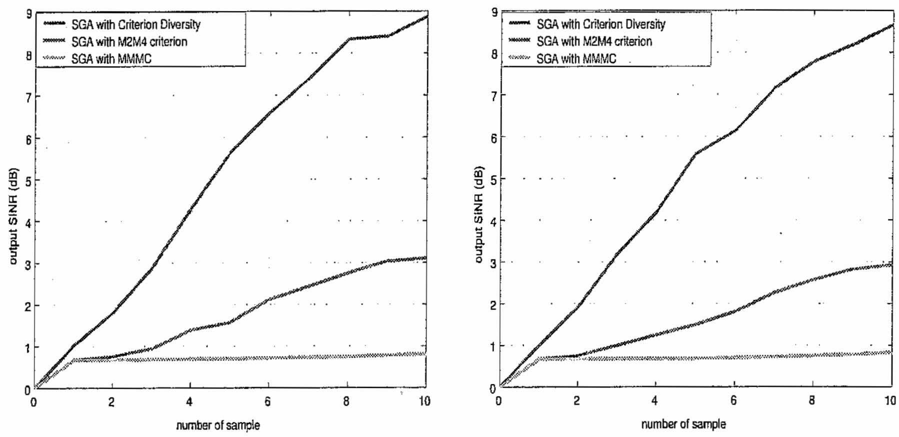

Figure 3.10: SGA convergence curves, SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

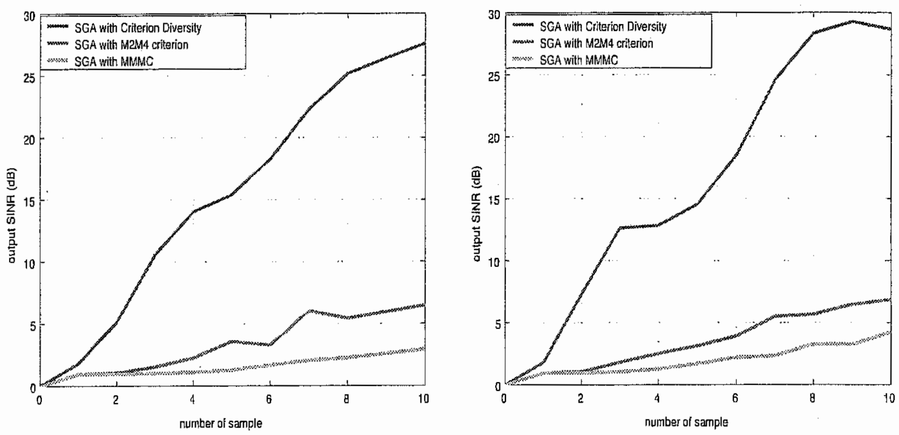

Figure 3.11: SGA convergence curves, SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

3.3 Sequential Random Algorithm with criterion diversity

3.3.1 Algorithm principle

In section 3.2, it has been shown that a gradient-based method has been developed to make the ESPAR antenna steers its beam and nulls automatically, where a sequential perturbation is needed to determine each component of the gradient vector. However, this technique requires M+1 measurements of the cost in each iteration which takes (M+1) times the duration of a conventional blind adaptive array processing.

One of the main algorithm for adaptive beamforming is called the random search algorithm, which consists of making a random change in the bias voltage vector simultaneously. This random search algorithm requires only a single measurement of the cost for each iteration. Nevertheless, nothing is learnt when a trial is completed because the next trial is completely independent from the previous one. Indeed it does not take any local continuity properties of the cost surface such as SGA.

That is why, a Sequential Random search Algorithm (SRA)has been proposed. In this one, a random change is made simultaneously in the reactance vector **x**. The cost (e.g., blind criteria: M2M4, MMMC or criterion diversity) is measured before and after the change, and these measures are compared. Then if the random change increases the cost, this one is accepted. Otherwise the change is rejected. According to [3], compared to pure random search algorithm, this one improves the performance of the adaptive ESPAR antenna.

In fact, let's denote $\mathbf{V} = [V_1, V_2, \cdots, V_M]$ the M-dimensional vector whose components are the bias voltages on reactances x_m, $m\epsilon[1, \cdots, M]$ respectively. Then, series of bias voltage vector $\mathbf{V}$(n) are generated :

$$\mathbf{V}(n) = \mathbf{R}(n), n\epsilon[1, \cdots, N] \tag{3.4}$$

where $\mathbf{R}$(n) is a vector of voltages values selected by a random number generator over the range of bias voltages on the varactors and n is the iteration step.

Each component of random vector $\mathbf{R}$(n) can be selected from different ways :

- random variables with uniform distribution over a $-b$ to b range (b is a positive value).
- gaussian sequence with zero mean and variance σ (σ is positive value).

In fact the value of b and σ may be constant but it seems more reasonable that the range of uniform distribution and the variance of gaussian distribution values decrease during the iteration processing. Therefore, according to [3] an alternative to the range parameter and the variance σ can be used as :

$$b(n) = \frac{b_0}{1 + \frac{n}{\tau . N}}, \tag{3.5}$$

$$\sigma(n) = \frac{\sigma_0}{1 + \frac{n}{\tau . N}}, \tag{3.6}$$

Similar methods are explained in [9] and [10] where parameters values decrease whereas the increment of the iteration step.

Consequently, if we denote J($\mathbf{V}$(n)) as an estimate of the cost value when $\mathbf{V}$(n)is fed in the reactances, this procedure can be algebraically described as :

$$\mathbf{V}(n+1) = \mathbf{V}(n) + \frac{1}{2}(1 + sgn[J(\mathbf{V}(n) + \mathbf{R}(n)) - J(\mathbf{V}(n))])\mathbf{R}(n), n = 1, 2, \cdots, N-1 \tag{3.7}$$

where $sgn[z]$ equals +1 for $z \geq 0$ and -1 for $z \leq 0$.

Then, this algorithm has been applied to criterion diversity. Consequently, the basic principle of the SRA using criterion diversity proceeds as follows :

- Step 1: Set initial values for the reactance vector $\mathbf{x}$. Generally $\mathbf{x}$ is set to a zero vector which leads to an omni-directional antenna.
- Step 2: Both blind criteria are computed (M2M4 and MMMC).
- Step 3: A random change is made on the voltage vector, which leads to a new reactance vector.
- Step 4: Calculate again values of the blind criteria and compute the difference before and after the update in the voltage vector.
- Step 5: The criterion which has the greatest difference is employed for this algorithm and then update the reactance vector. Otherwise if the differences of the two criteria are negative, the change is rejected and the algorithm keeps the previous reactance vector. Finally go back to step 2 and repeat the process during N iterations.

The flowchart of sequential random algorithm using criterion diversity is depicted in Figure 3.12 .

3.3.2 Simulation results

Nevertheless, according to [3] simulations have been conducted with different forms of random vectors $\mathbf{R}$ and it has been shown that gaussian distribution provides better results than uniform distribution. Consequently, gaussian distribution will be used for all matlab simulations. Furthermore, iteration number will be set to 50 for all SRA simulations.

First, effect of the σ_0 parameter has been studied. Figure 3.13 represents the result of SRA simulations using only MMMC criterion with different values of the σ parameter. In this one DoA, SNR andτ parameters values are fixed.

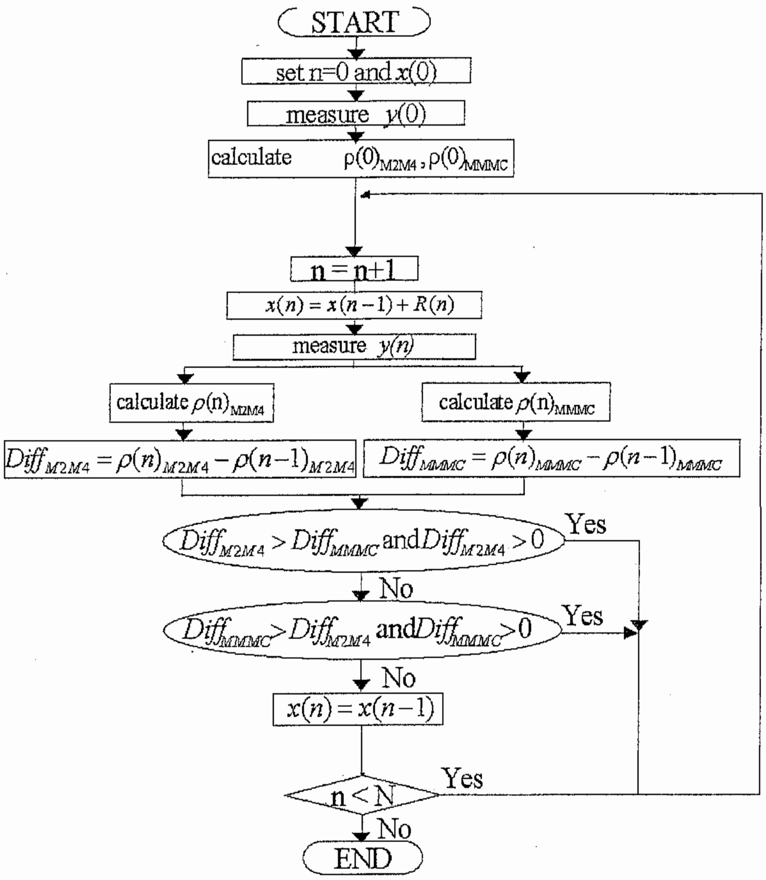

Figure 3.12: SRA with criterion diversity flowchart.

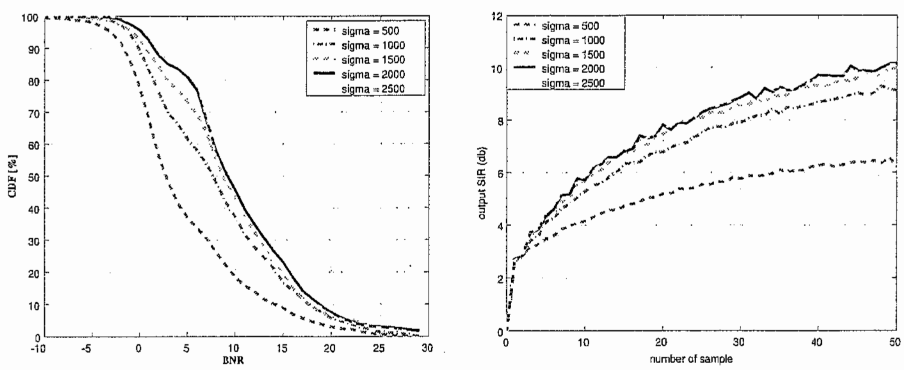

Figure 3.13: SRA CDF and convergence curves with different sigma parameter values, SNR= 10dB, DoA [0° 45°].

Then the effect of the τ parameter has been considered in the case of the gaussian distribution of the random vector $\mathbf{R}$. In figure 3.14, CDF and convergence curves are depicted whereas τ parameter value is changed. These two curves show that the SRA seems to be more efficient if the τ parameter equals 5. Consequently, in all the sequential random algorithm simulations τ parameter value is set to 5.

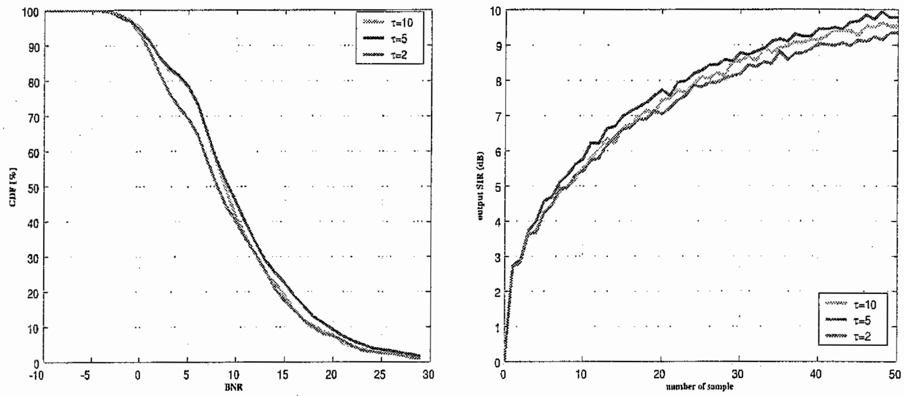

Figure 3.14: SRA CDF and convergence curves with different τ parameter values, SNR= 10dB, DoA [0° 45°].

Furthermore, figures 3.15, 3.16 and 3.17 are results of SRA adaptive beamforming while using only M2M4 criterion, MMMC or criterion diversity. In fact, simulations have been conducted for same SNR values (5dB, 10dB and 20dB)and DoA's ([0° 45°] and [0° 60°]) as the steepest gradient algorithm.

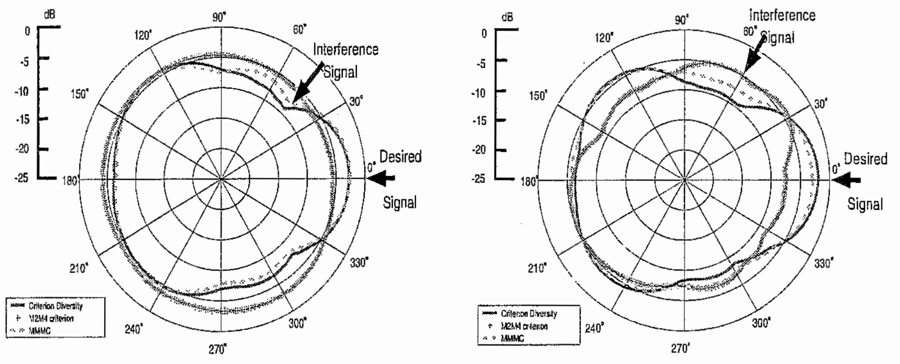

Figure 3.15: SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

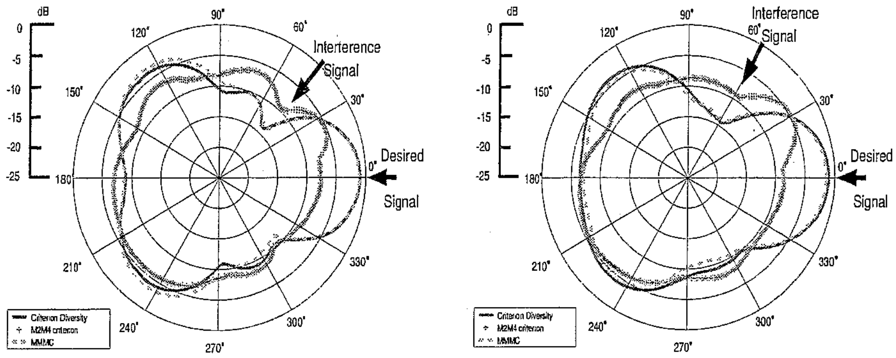

Figure 3.16: SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

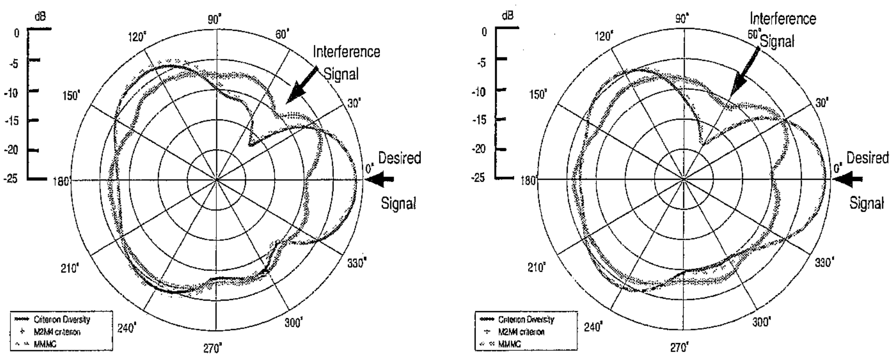

Figure 3.17: SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

Figures 3.18, 3.19 and 3.20 are convergence curves in order to show efficiency of SRA criterion diversity. In fact, it compares output SIR with average of 1000 trials of 50^{th} iterations for SRA.

Finally, it appears that same as steepest gradient algorithm, whatever are the initial SNR and DoA's values criterion diversity leads to a better blind beamforming when it is applied to the sequential random algorithm.

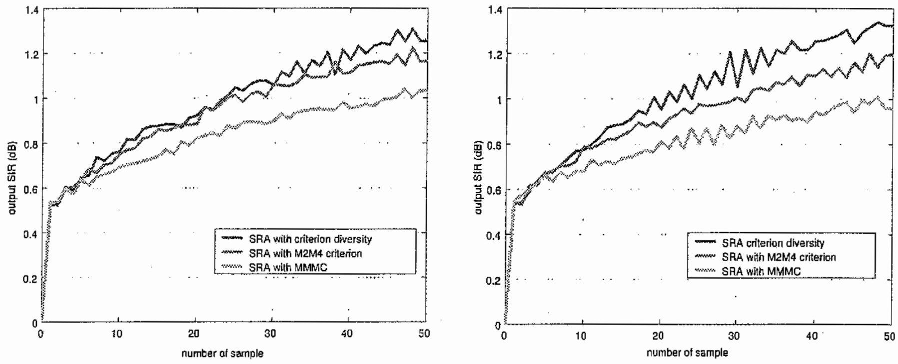

Figure 3.18: SRA convergence curves, SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

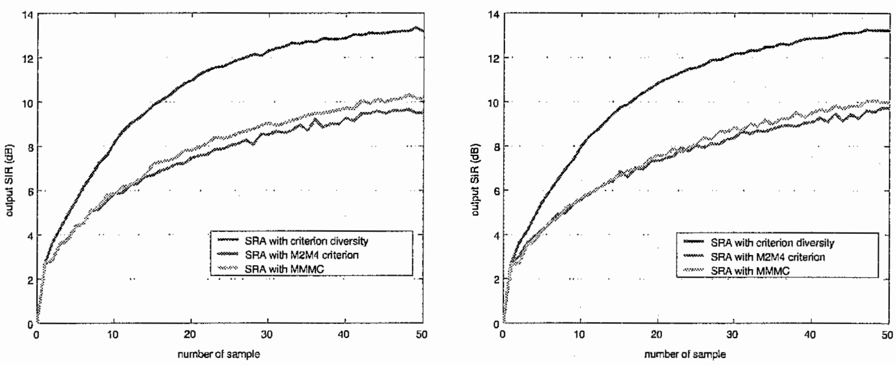

Figure 3.19: SRA convergence curves, SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

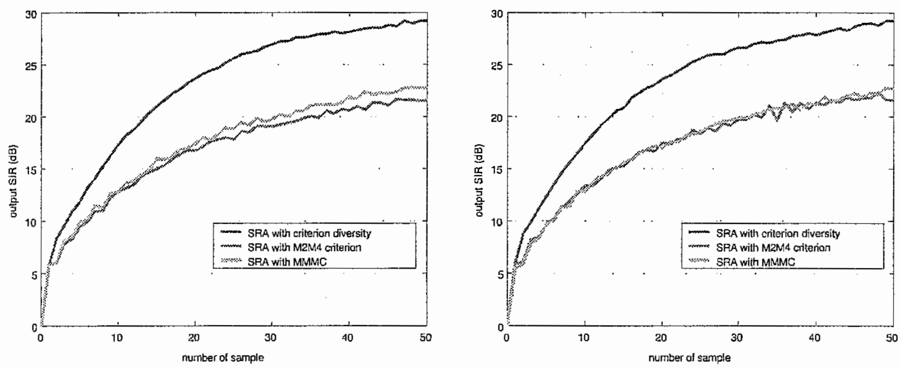

Figure 3.20: SRA convergence curves, SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

3.4 Comparison of the two algorithms with criterion diversity

Now that we have described both algorithms using criterion diversity, we can discuss about the advantages and the drawbacks of each algorithm.

First, we can notice that beams patterns and convergence curves have shown that adaptive algorithms which employ criterion diversity lead to a better blind adaptive beamforming. Then, Table 3.2 shows characteristics for both blind adaptive algorithms appled to criterion diversity for ESPAR antennas.

Table 3.2: Algorithms with criterion diversity for ESPAR antennas

Algorithms	SGA	SRA
Speed	Low	High
Stability	High	Low

In fact, simulations have shown that when the value of the Signal-to Noise Ratio (SNR) is high (around 20 dB), the accuracy of SRA using criterion diversity is better and forms a beam where the desired signal is and a deep null towards the interference signal . On the contrary when SNR is low, SGA becomes much more efficient than the SRA. That is why, an adaptive control can be imagined by using both algorithms with criterion diversity because as we usually can not know the value of SNR, both of them are efficient in different conditions.

Furthermore, time duration problems can be encountered. In fact, the SGA algorithm shows more complexity, thus takes much more time to be processed than the SRA technique if the iteration number is the same for these two algorithms. It can be explained by the fact that SGA has to compute a gradient vector for each parasitic element for a 7-element ESPAR antenna (see Fig. 4).

Although SGA can form both a beam and nulls for patterns even if the SNR value is high, complexity of the SGA is much higher than the SRA, that is why the SRA is better to use for high SNR. It is still also more efficient than SRA in certain conditions when the desired and interference signals have almost the same power.

Chapter 4

Diversity of both algorithms and criteria

4.1 Adaptive control principle

The main objective of this adaptive control is to combine the SGA with the SRA, each of these algorithms using criterion diversity. Consequently, adaptive control should improve patterns whatever the power value of the impinging signal is. In fact, the main ability of diversity of both algorithms and criteria is to be able to switch from the SRA to the SGA during the blind beamforming.

Basically, this algorithm proceeds as follows :

- Step 1: Process the Sequential Random Algorithm with criterion diversity during N_{SNR} times.
- Step 2: Proceed to a SNR estimation after N_{SNR} iterations.
- Step 3: Compare SNR to a threshold value (set to be 10dB).
- Step 4: If SNR is below the threshold value, process the SGA algorithm with criterion diversity during (N-N_{SNR}) times. Otherwise process again the SRA algorithm during (N-N_{SNR}) times.

There are two other different parameters which control this adaptive algorithm: N_{SNR} and also the threshold value. Because iteration number has been set to 50, N_{SNR} value will be set to half the processing of the algorithm (N_{SNR} set to 25).The flowchart of this adaptive control is depicted in Figure 4.1.

Consequently, if after N iterations SNR is higher than the threshold value, adaptive control will be at least as efficient as the Sequential Random algorithm using criterion diversity. On the contrary, if SNR estimation is still lower than the threshold value after N_{SNR} iterations, it means that the SGA blind beamforming is better than the SRA, that is why the adaptive control switches from the SRA to the SGA.

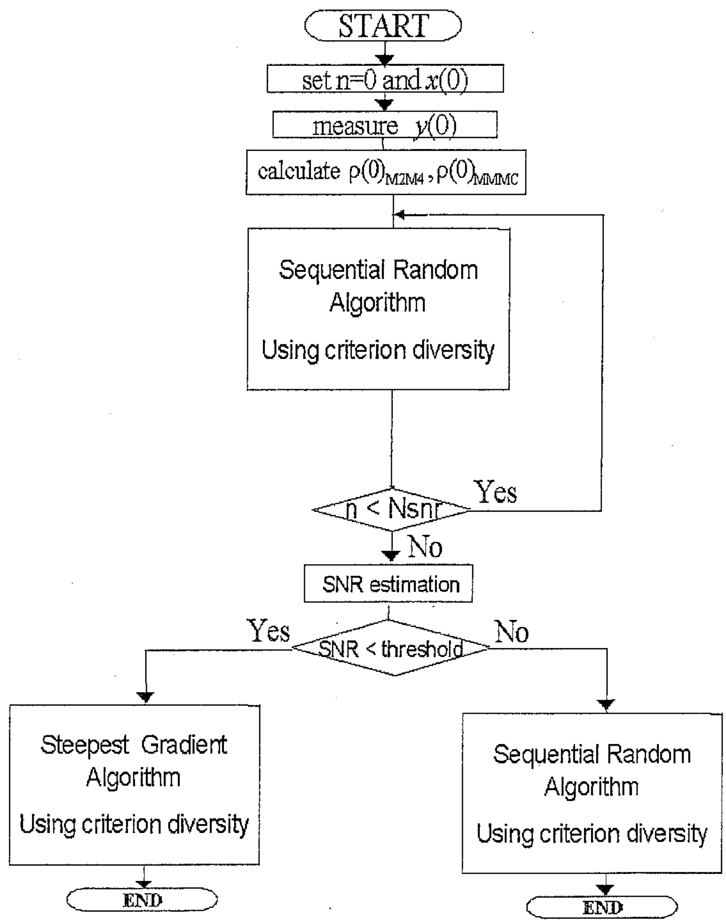

Figure 4.1: Adaptive Control using concurrent algorithm and criterion diversity.

4.2 Simulations and discussions

Simulations have been conducted in order to evaluate the performance of the adaptive control of the algorithms. ESPAR antenna patterns have been simulated for several DoA's and for different SNR values (between 0 and 20dB). In the simulations, a desired signal is impinging on the ESPAR antenna and furthermore an interference signal is also added.

Figures 4.2, 4.3 and 4.4 are beamforming results of adaptive control using concurrent criterion and algorithm diversity. It shows that whatever the value of the SNR is, blind beamforming of the adaptive control will be better than the SGA and the SRA using criterion diversity.

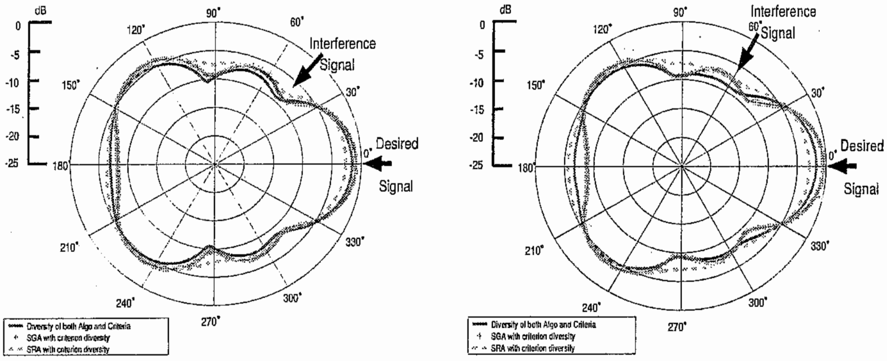

Figure 4.2: SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

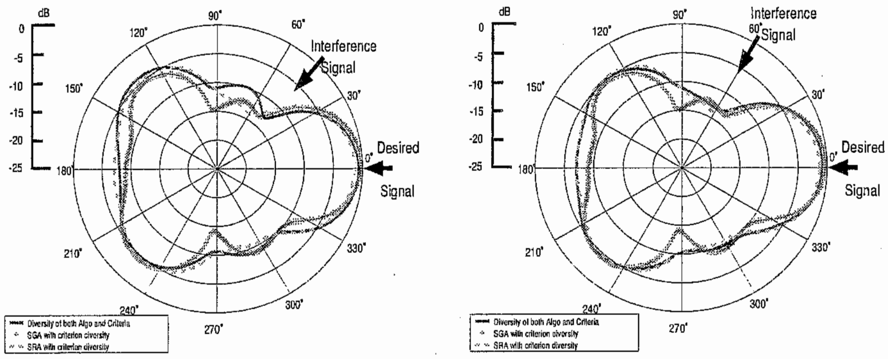

Figure 4.3: SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

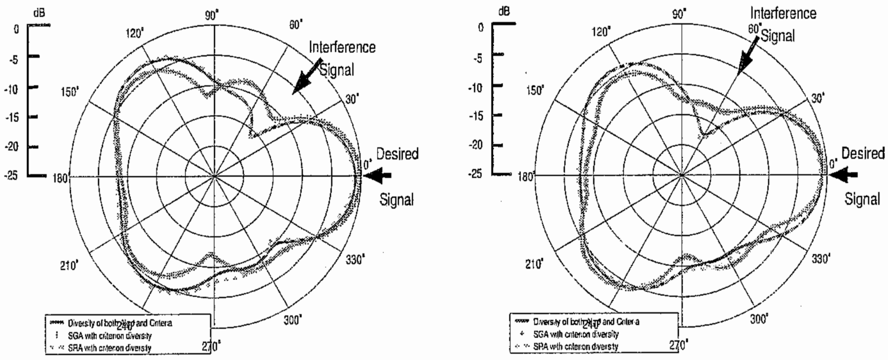

Figure 4.4: SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

Figures 4.5, 4.6 and 4.7 are convergence curve swhich compares concurrent criterion and algorithm diversity versus the SGA and the SRA with criterion diversity. These simulations have been conducted with 1000 trials of

50^{th} iterations (N parameter) and N_{SNR} is set to 25. It shows that during N_{SNR}^{th} first iterations, convergence curves of concurrent criterion and algorithm diversity and SRA are almost the same. But in fact, for every sample at the N_{SNR}^{th} iteration, if SNR value is below the threshold parameter, SGA algorithm will be used instead of the SRA.

As threshold value is set to 10dB, few samples of concurrent criterion and algorithm diversity will switch from the SRA to the SGA. In this case, reactance vector will be initialized, that is why output SIR value of concurrent criterion and algorithm diversity seems to drop and then after converge quickly to reach output SIR value of SGA algorithm.

Finally, concurrent criterion and algorithm diversity is as efficient as the SGA algorithm (for this simulation, SGA convergence curve is better than SRA). However, concurrent criterion and algorithm diversity is less complex than SGA (because SRA is less complex than SGA), that is why this new algorithm is more efficient than others.

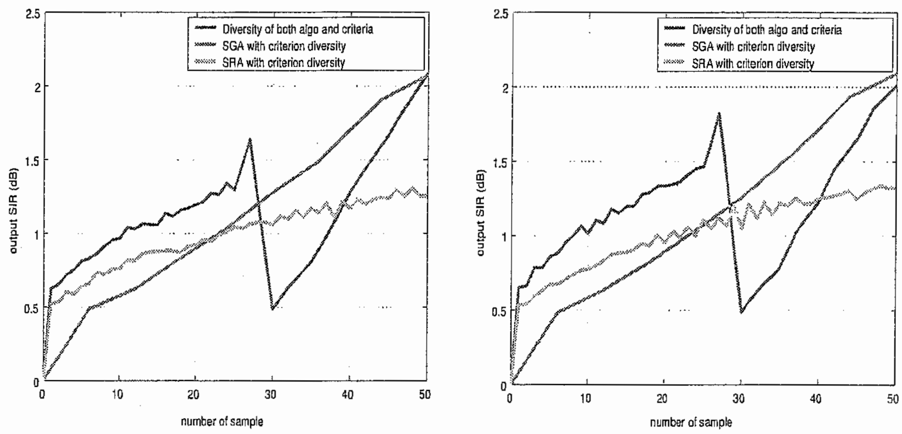

Figure 4.5: SNR= 5dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

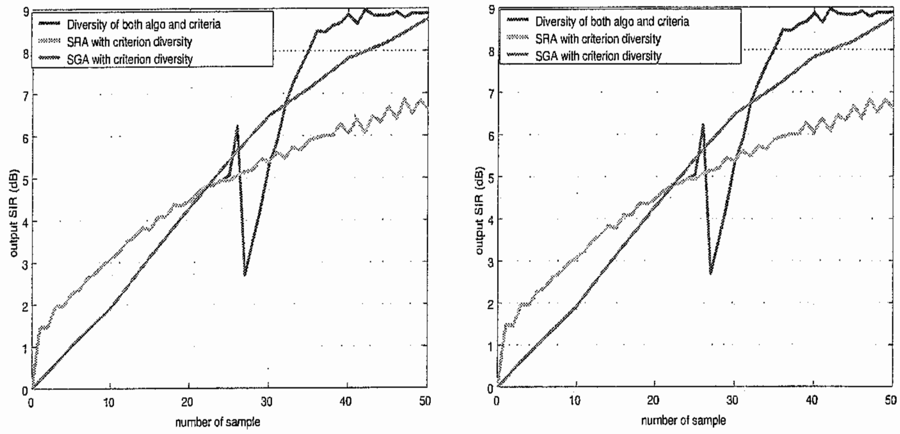

Figure 4.6: SNR= 10dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

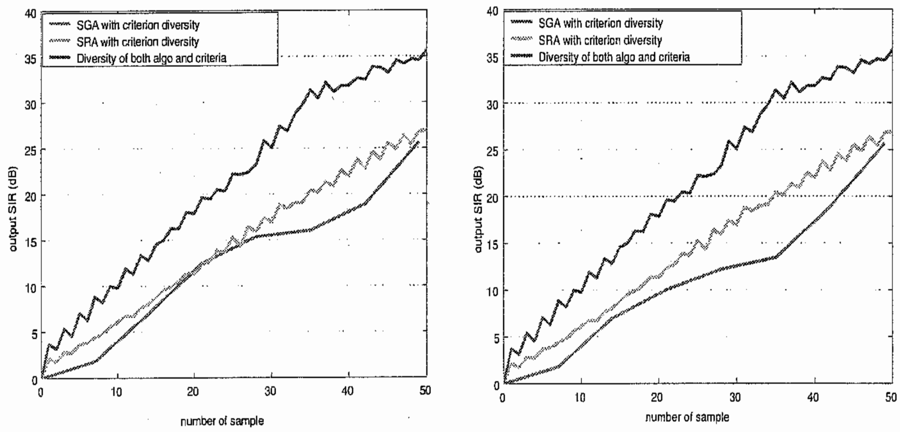

Figure 4.7: SNR= 20dB, DoA [0° 45°] on the left and DoA [0° 60°] on the right.

Conclusion

In this paper, a new solution to the deployment of smart antennas for wireless communications systems is presented. First, basic principle of criterion diversity has been developed. Furthermore this technique has been used for two different criteria : M2M4 and MMMC. These blind criteria are complementary because the first one has a fast convergence but a low stability which are the exactly opposite characteristics of the MMMC.

Then, this method has been applied for two different blind adaptive algorithms: the Steepest Gradient Algorithm (SGA) and the Sequential Random Algorithm (SRA). Both of these adaptive controls have been adapted for using criterion diversity. Simulations have been conducted in order to compare them when they are using only one criterion (M2M4 or MMMC) or criterion diversity.

For both of them, it has been shown that criterion diversity leads adaptive algorithms to form faster directional patterns with a more stable convergence. However, efficiency of both SGA and SRA depend on SNR conditions (on signal power). In fact, a blind adaptive control could be imagined while using both of SGA and SRA : this is the principle of using concurrent criterion and algorithm diversity.

Consequently, an adaptive control has been performed, which can use both of these blind adaptive algorithms. Therefore, simulations have been conducted in order to validate this algorithm.

That is why, using diversity of both algorithms and criteria leads to improvement in blind adaptive beamforming for ESPAR antennas.

During my internship, I have submitted two papers for papers for conferences. These two papers called "Concurrent Criterion - and Algorithm - Diversity Blind Adaptive Beamforming for an ESPAR antenna" have been published in IEICE national conference.

Acknowledgements

I would like to express my gratitude to Mr. Tomoyuki Aono, who was an excellent supervisor, Dr. Takashi Ohira, head of Department 3, who was a very good head supervisor, and Dr. Bokuji Komiyama, president of ACR for their helpful encouragements and their interests to my work.

I especially would like to thank Mr. Eddy Taillefer who introduced me to ACR laboratories and with whom I had very interesting and helpful discussions for my work.

I would like to thank all the ACR department 3 team, Mr. Tomoyuki Aono, Dr. Qing Han, Mr. Brett Hanna, Mr. Akifumi Hirata, Dr. Thomas Hunziker, Mr. Kyoichi Iigusa, Mr. Takuma Sawaya, Mr. Tatsuya Shimizu, Mr. Eddy Taillefer, Mr. Hiroki Tanaka and Mr. Satoru Tawara for their collaboration and discussions and for having good times with them.

I would like to express my many thanks to Ms. Naoko Nakaya, Ms. Mami Hamba and Ms. Natsuko Harada for their help to fill in the numerous formality papers.

I would like to thank the SHIEN group and especially Ms. Yukiko Ishikawa and Ms. Makiko Tatsumi for the their fantastic help to prepare all the procedures to come in Japan and for their kind personality.

I finally express my gratitude to all ATR staff for being very nice and friendly.

Bibliography

[1] T. Ohira, "Renaissance Of Analog Beamforming Approach To Adaptive Array Antennas", *ISSSE 2001*, WE3-B1, July 2001.

[2] J. Cheng, Y. Kamiya and T. Ohira, "Adaptive Beamforming of ESPAR Antenna Based on Steepest Gradient Algorithm", *IEICE Trans. commun.*, Vol. E84-B, No. 7, July 2001.

[3] J. Cheng, A. Hirata, B. Shishkov and T. Ohira, "Sequential Random Search Algorithm for Adaptive Beamforming of ESPAR antenna", *Technical Report of IEICE*, A.P2001-107, RCS2001-146 Oct.2001.

[4] R. Matzner and K. Letsch, "SNR estimation and blind equalization (deconvolution) using the kurtosis,", *Proc. of IEEE IMS Workshop on Information Theory Statistics*, Alexandria, VA, p.68, Oct. 1994.

[5] T. Ohira, "Blind aerial beamforming based on a higher-order maximum moment criterion(Part I : Theory)," *Proc. of Asia-Pacific Microwave Conference 2002*, pp.652-655, San Antonio, TX, June 2002.

[6] K. Takizawa, T. Aono and T. Ohira "Criterion Diversity: A New Blind Adaptive Beamforming Scheme for ESPAR Antennas".*ECWT 2003* in press.

[7] K. Gyoda and T.Ohira, "Design of electronically steerable passive array radiator (ESPAR) antenna," 2000 IEEE Antennas and Propagation Society International Symposium, Transmitting Waves of Progress to the Next Millenium, pp.922, Salt Lake City, Utah, July 16-21,2000.

[8] A. Akiyama, K. Gyoda, T. Ohira and M. Ando, "Basic study of Espar antenna - Simultaneous forming of a beam and multpile nulls" IEICE Technical Report, AP2000-128, SANE2000-105, SANE2000-105, RCS2000-151, Niigata, Japan, oct. 2000.

[9] B. Shishkov, J. Cheng and T. Ohira, "Adaptive beamforming of ESPAR antenna - non-conventional approach,"2001 Inter. Symposium on Signals, Systems, and Electronics (ISSSE'01), pp. 412-415, July 24-27, 2001, Tokyo, Japan.

[10] B. Shishkov, J. Cheng and T. Ohira, "Non-conventional adaptive beamforming of ESPAR antenna", 2001 Asia-Pacific Radio Science Conference (AP-RASC'01), p. 298, Aug. 1-4, 2001, Tokyo, Japan.

Appendix : MatLab Source Files

```
BothDiversities.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%        Algo de Aono       %
%        Version Finale     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all % new workspace

m=6;                   % number of parasitic elements
z_s=50;                % output impedance of transmitter
Ns=64;                 % number of samples
M=4;                   % MPSK
N=50;                              % Iteration Number
Nsnr=25;               % switching number
SNR_dB=20;             % Initial SNR (in dB)
n0=10^(-SNR_dB/10);    % noise power
int_type=2;            % Signal type (1:QPSK, 2:RAND-PHASE PSK, 3:NO INTERFERENCE)
step_s=1500;               % step size
rep_sig=exp(-j*([0:M-1]/M*2*pi));

%%%%% SRA parameter %%%%%%
vol_size = [-2048 2047];
SIGMA = [2000];
%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%% SGA parameter %%%%%%
delta_x=32;                % perturbation size
criterion=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%

X_size = -(0.0217*vol_size+49.21);

threshold = 10;      % threshold parameter for switching algorithms (in dB)


% admittance matrix components: 6 components for a 7 element ESPAR antenna

y00=0.0008616-j*0.0120795;   % y00 parameter
y10=-0.0006963+j*0.0036462;  % y10 parameter
y11=0.0044216-j*0.0071600;   % y11 parameter
y21=0.0009721+j*0.0047851;   % y21 parameter
y31=-0.0005376-j*0.0011297;  % y31 parameter
y41=0.0001701-j*0.0002950;   % y41 parameter

%%%%%%%%%%%%%%%%%%%%%%%
% Y : admittance matrix
Y=[y00 y10 y10 y10 y10 y10 y10;
    y10 y11 y21 y31 y41 y31 y21;
    y10 y21 y11 y21 y31 y41 y31;
    y10 y31 y21 y11 y21 y31 y41;
    y10 y41 y31 y21 y11 y21 y31;
```

```
y10 y31 y41 y31 y21 y11 y21;
y10 y21 y31 y41 y31 y21 y11];

% inverse of the admittance matrix : Z
Z=inv(Y);

%initialisation
range =300;
CDF = zeros(1,range);   %CDF
CDF_SINR = zeros(1,range);
sample =1000;       % number of samples used for the simulation

% DoA: Direction of Arrival=> 1st number: desired signal, others: interferences signals
DoA=[0 45]
ret=[];
ga =[];

Tablo=zeros(1,N);
Tablo_SINR=zeros(1,N+1);
Tablo_SINR_half=zeros(1,Nsnr+1);
SINR=zeros(1,N+1);
Tab_BNR_CDF=zeros(1,sample);
Tab_BNR_BestCriterion=zeros(1,sample);
resolution=360;
Tablo_Gain=zeros(1,resolution);
cptSGA=0;
cptSRA=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Time display
start_p = time_disp1(sample);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% beginning of the sample loop
for ii = 1:sample
    vol_size = [-2048 2047];
    X_size = -(0.0217*vol_size+49.21);

    %%%%%%%%%%%%%%%%%%%%%%%%
    % X : Reactance Matrix
    vol = randint(m,1,vol_size);
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);

    % Unit Vector
    u_0=zeros(m+1,1);
    u_0(1)=1;

    phi=DoA/360*2*pi;               % [deg->rad]
    n_arrival=length(phi);
```

```
power_dB = [3];
power_i=10^(-power_dB/10);
power=[1 power_i];

%%%%%%%%%%%%%%%%%%%%%%%%%
% a_phi : steering vector
a_phi=zeros(m+1,n_arrival);
for i=1:n_arrival
    a_phi(:,i)=[1 exp(j*pi/2*cos(phi(i)-[0:m-1]/m*2*pi))]';
end
%%%%%%%%%%%%%%%%%%%%%%%%%
% Vectors Initialization
J_tr=zeros(1,N);
dJ_dx=zeros(1,m+1);
dJ_dx_m2m4=zeros(1,m+1);         % DJ_dx for m2m4
dJ_dx_mmmc=zeros(1,m+1);         % DJ_dx for mmmc
Tab_J_m2m4=zeros(1,N);           % Table of gradient vector for m2m4 criterion
Tab_J_mmmc=zeros(1,N);           % Table of gradient vector for mmmc criterion
Tab_J_CriterionDiversity=zeros(1,N); % table of gradient vector for criterion diversity
Tab_Criterion=zeros(1,N);
SER=zeros(1,N);
sinr=zeros(1,N+1);
sinr_half=zeros(1,Nsnr+1);
grad_J=zeros(1,N);

%%%%%%%%%%%%%%%%%%%%%%%%%
% Signal modulation + interference + noise
[sk,phi1,nn]=mod_CCK(Ns);
[ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

Tab_J=zeros(1,N);          % Recording the value of J
Tab_BNR=zeros(1,N);        % Recording the value of the BNR
gain_dBMin=0;

sinrMax = 0;                     % Recording the sinr maximum value.
J_pre_m2m4=m2m4(y,M);         % best criterion
J_pre_mmmc=mmmc(y,M);         % best criterion

%%%%%%%%%%%%%%%%%%%%%%%%%
% w : Weight vector

w=2*X(1,1)*inv(Z+X)*u_0;
a=conj(w')*ref;
sinr(1)=mean(abs(a).^2)/mean(abs(y-a).^2); % SINR
sinr_half(1)=mean(abs(a).^2)/mean(abs(y-a).^2); % SINR

Tablo(1)=Tablo(1)+(1/sample)*sinr(1);

if (sinrMax<sinr(1))
    sinrMax = sinr(1);
```

```
end
n = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 SRA               %
%    with criterion diversity       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while n < (Nsnr)

    n = n+1;
    %sigma parameter
    sigma=round(SIGMA/(1+n/250));
    volshift=randn(m,1)*sigma;  % for gaussian distrubution
    volold=vol;
    vol=vol+volshift;
    for index=1:m                % -REAMax<=rea<=REAMax
        if vol(index) > vol_size(2)
            vol(index) = vol_size(2);
        elseif vol(index)<vol_size(1)
            vol(index)=vol_size(1);
        end
    end
    % reactance matrix
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);
    % ESPAR antenna output formulation
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    %blind criteria
    J_m2m4=m2m4(y,M);
    J_mmmc=mmmc(y,M);

    % SINR
    w=2*X(1,1)*inv(Z+X)*u_0;     % weight vector
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);
    sinr_half(n)=mean(abs(a).^2)/mean(abs(y-a).^2); %SINR

    % criterion selection
    Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
    Diff_mmmc=(J_mmmc)-(J_pre_mmmc);

    if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
        % m2m4 criterion choosed
        J_pre_m2m4 = J_m2m4;
        if n == N
            break
        end
    elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
```

```
        % mmmc criterion choosed
        J_pre_mmmc = J_mmmc;
       if n == N
            break
        end
    else
        vol=volold;
        if n == N
            break
        end
        n = n+1;
        vol=vol-volshift;
        % for every parasitic elements
        for index=1:m               % -REAMax<=rea<=REAMax
            if vol(index) > vol_size(2)
                vol(index) = vol_size(2);
            elseif vol(index)<vol_size(1)
                vol(index)=vol_size(1);
            end
        end
        X = -(0.0217*vol+49.21);
        X = diag([z_s;X*j]);
        % ESPAR output formulation
        [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

        % blind criteria
        J_m2m4=m2m4(y,M);
        J_mmmc=mmmc(y,M);
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        % SINR
        w=2*X(1,1)*inv(Z+X)*u_0;
        a=conj(w')*ref;
        sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);
        sinr_half(n)=mean(abs(a).^2)/mean(abs(y-a).^2);

        % criterion selection
        Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
        Diff_mmmc=(J_mmmc)-(J_pre_mmmc);
        if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
            J_pre_m2m4 = J_m2m4;
            if n == N
                break
            end
        elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
            J_pre_mmmc = J_mmmc;
            if n == N
                break
            end
```

```
        else
            vol=volold;
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %X = -(0.0217*vol+49.21);
            %X = diag([z_s;X*j]);
            %[sk, phi1, nn]=mod_CCK(Ns, phi1, nn);
            %[ref, y]=esper_out(sk, a_phi, Z, X, n0, power, Ns, M, m, int_type);
            %w=2*X(1, 1)*inv(Z+X)*u_0;
            %a=conj(w')*ref;
            %sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);
            %sinr_half(n)=mean(abs(a).^2)/mean(abs(y-a).^2);
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            if n == N
                break
            end
        end
    end
end
% steering vector
resolution=360;
phi_d=linspace(0, 2*pi*(1-1/resolution), resolution);
ar=ones(m+1, resolution);
for i=2 :m+1
    ar(i, :)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));

end
% reactance matrix
X = -(0.0217*vol+49.21);
X = diag([50;X*j]);
w=2*X(1, 1)*inv(Z+X)*u_0;
arout=abs(w'*ar);

sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2); %SINR
sinr_half(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2);
gain_dB = 20*log10(arout/max(arout));
BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
Tab_BNR(ii)=BNR;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%    Half of the Aono Algorithm    %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for (y=1 : 360)
    if (gain_dB(y) < gain_dBMin)
        gain_dBMin=gain_dB(y);
    end
end
if ((sinr(n))<threshold)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %          SGA with criterion diversity        %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
vol_size = [2048 -2047];
X_size = -(0.0217*vol_size+49.21);
fprintf('SGA algorithm choosed¥n')
vol_SGA = zeros(1,m+1);
X = diag(-(0.0217*vol_SGA+49.21)*j);
X(1,1)=z_s;
n=(Nsnr+1);
cptSGA=cptSGA+1; % SGA counter
while n<N
    n=n+1;
    % ESPAR signal modelisation
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);
    % blind criteria
    J_pre_m2m4=m2m4(y,M);
    J_pre_mmmc=mmmc(y,M);

    if criterion==1
        J_pre=J_pre_m2m4;
    else
        J_pre=J_pre_mmmc;
    end

    % w : Weight vector
    w=2*X(1,1)*inv(Z+X)*u_0;
    a=conj(w')*ref;

    resolution=360;
    phi_d=linspace(0,2*pi*(1-1/resolution),resolution);
    ar=ones(m+1,resolution);

    for i=2 :m+1
        ar(i,:)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
    end

    arout=abs(w'*ar);
    gain_dB = 20*log10(arout/max(arout));
    gain=(arout/max(arout));
    Tab_BNR(n)  = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);

    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2); % SINR
    Tablo(n)=Tablo(n)+(1/sample)*sinr(n);
    if (sinrMax<sinr(n))
        sinrMax = sinr(n);
    end

    X_test=X;
    % for every parasitic element
    for i=2:m+1
```

```
        X_test(i,i)=X_test(i,i)+j*delta_x;
        if imag(X(i,i)) > X_size(2)
            X(i,i) = X_size(2)*j;
        end
        if imag(X(i,i))<X_size(1)
            X(i,i)=X_size(1)*j;
        end

        % ESPAR signal modelisation
        [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        [ref,y]=esper_out(sk,a_phi,Z,X_test,n0,power,Ns,M,m,int_type);
        % compute 2 criteria's values
        J_m2m4=m2m4(y,M);
        J_mmmc=mmmc(y,M);
        dJ_dx_m2m4(i)=(J_m2m4-J_pre_m2m4)/delta_x;
        dJ_dx_mmmc(i)=(J_mmmc-J_pre_mmmc)/delta_x;

        X_test(i,i)=X(i,i);
    end
    % compute criteria's norms
    Norm_dJ_dx_m2m4=norm(dJ_dx_m2m4);
    Norm_dJ_dx_mmmc=norm(dJ_dx_mmmc);

    % criterion selection
    if (Norm_dJ_dx_m2m4>Norm_dJ_dx_mmmc)
        criterion=1;
        dJ_dx=dJ_dx_m2m4;
    else
        criterion=0;
        dJ_dx=dJ_dx_mmmc;
    end

    X=X+diag(j*step_s*dJ_dx);% update of the reactance matrix
    for i=2:m+1
        if imag(X(i,i)) > X_size(2)
            X(i,i) = X_size(2)*j;
        end
        if imag(X(i,i))<X_size(1)
            X(i,i)=X_size(1)*j;
        end
    end
    % Recording of the values of Norm(m2m4) and Norm(mmmc)
    Tab_Norm_m2m4(n)=Norm_dJ_dx_m2m4;
    Tab_Norm_mmmc(n)=Norm_dJ_dx_mmmc;
    if criterion==1
        J_tr(n)=J_pre_m2m4;
        grad_J(n)=Norm_dJ_dx_m2m4;
    else
        J_tr(n)=J_pre_mmmc;
        grad_J(n)=Norm_dJ_dx_mmmc;
```

```
    end
    Tab_Criterion(n)=criterion; % updating the table
end

resolution=360;
phi_d=linspace(0,2*pi*(1-1/resolution),resolution);
ar=ones(m+1,resolution);

for i=2 :m+1
    ar(i,:)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
end

w=2*X(1,1)*inv(Z+X)*u_0; % weight vectors

% ESPAR signal modelisation
[sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
[ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

% compute 2 criteria's values
J_pre_m2m4=m2m4(y,M);
J_pre_mmmc=mmmc(y,M);

if criterion==1
    J_pre=J_pre_m2m4; % m2m4 criterion choosed
else
    J_pre=J_pre_mmmc; % mmmc choosed
end
J_tr(n+1)=J_pre;
a=conj(w')*ref;
sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2); % SINR
Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
if (sinrMax<sinr(n))
    sinrMax = sinr(n);
end

arout=abs(w'*ar);
gain_dB = 20*log10(arout/max(arout));
gain=(arout/max(arout));
for (y=1 : 360)
    if (gain_dB(y) < gain_dBMin)
        gain_dBMin=gain_dB(y);
    end
end

Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii;

BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
Tab_BNR(ii)=BNR;
ret = [ret ; BNR];
cnt = round(BNR);
```

```
        cnt_s = round(sinr(end));
        SINR=(SINR*(ii-1)+sinr)/ii;
    else
        cptSRA=cptSRA+1; % increments SRA counter
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %              SRA ALGO choosed          %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        while n < N
            n = n+1;
            % random change
            sigma=round(SIGMA/(1+n/250));
            volshift=randn(m,1)*sigma;  % for gaussian distrubution
            volold=vol;
            vol=vol+volshift;
            for index=1:m              % -REAMax<=rea<=REAMax
                if vol(index) > vol_size(2)
                    vol(index) = vol_size(2);
                elseif vol(index)<vol_size(1)
                    vol(index)=vol_size(1);
                end
            end

            X = -(0.0217*vol+49.21);
            X = diag([z_s;X*j]);
            % ESPAR signal modelisation
            [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
            [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

            % compute both blind criteria
            J_m2m4=m2m4(y,M);
            J_mmmc=mmmc(y,M);
            w=2*X(1,1)*inv(Z+X)*u_0;      % weight vector
            a=conj(w')*ref;
            sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);  % SINR
            % compute criteria's difference
            Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
            Diff_mmmc=(J_mmmc)-(J_pre_mmmc);

            if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
                % m2m4 choosed
                J_pre_m2m4 = J_m2m4;
                if n == N
                    break
                end
            elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
                % mmmc choosed
                J_pre_mmmc = J_mmmc;
                if n == N
                    break
                end
```

```
else
    vol=volold;
    if n == N
        break
    end
    n = n+1;
    vol=vol-volshift;
    for index=1:m                    % -REAMax<=rea<=REAMax
        if vol(index) > vol_size(2)
            vol(index) = vol_size(2);
        elseif vol(index)<vol_size(1)
            vol(index)=vol_size(1);
        end
    end
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);
    % ESPAR signal modelisation
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    % compute blind criteria
    J_m2m4=m2m4(y,M);
    J_mmmc=mmmc(y,M);

    w=2*X(1,1)*inv(Z+X)*u_0;
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);% SINR
    % compute blind criteria's difference
    Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
    Diff_mmmc=(J_mmmc)-(J_pre_mmmc);
    if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
        % m2m4 choosed
        J_pre_m2m4 = J_m2m4;
        if n == N
            break
        end
    elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
        % mmmc choosed
        J_pre_mmmc = J_mmmc;
        if n == N
            break
        end
    else
        % none of change increases the cost
        vol=volold;
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %X = -(0.0217*vol+49.21);
        %X = diag([z_s;X*j]);
        %[sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        %[ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);
```

```
                %w=2*X(1,1)*inv(Z+X)*u_0;
                %a=conj(w')*ref;
                %sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                if n == N
                    break
                end
            end
        end
    resolution=360;
    phi_d=linspace(0,2*pi*(1-1/resolution),resolution);
    ar=ones(m+1,resolution);
    for i=2 :m+1
        ar(i,:)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
    end
    % reactance matrix
    X = -(0.0217*vol+49.21);

    X = diag([50;X*j]);
    w=2*X(1,1)*inv(Z+X)*u_0;
    arout=abs(w'*ar);
    sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2);
    Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
    % CDF
    gain_dB = 20*log10(arout/max(arout));
    Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii;
    BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
    Tab_BNR(ii)=BNR;
   end
 end
cptSGA
cptSRA
```

espar_formulation.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     ESPAR antenna output     %
%         formulation          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ref, y]=esper_out4(uk, a_phi, Z, X, n0, power, Ns, M, m, sig_type, f_o)

[aaa, n_arrival]=size(a_phi);

% unit vector
u_0=zeros(m+1, 1);
u_0(1)=1;

for i=1 :n_arrival
    if i==1
        % CCK modulation
        xc=a_phi(:, i)*uk*sqrt(power(i));
        ref=xc;
    else
        % Interference type
        if sig_type==1
            % QPSK SIGNAL modelisation
            xc=xc+a_phi(:, i)*sqrt(power(i))*exp(j*2*pi*randint(1, Ns, M)/M);
        elseif sig_type==2
            % RAND-PHASE SIGNAL modelisation
            xc=xc+a_phi(:, i)*sqrt(power(i))*exp(j*randint(1, Ns, 360)/360*2*pi);
        end
    end
end
% weight vector  (w=w_I+j*w_Q)
w=2*X(1, 1)*inv(Z+X)*u_0;
% ESPAR antenna output formula
y=conj(w')*xc+sqrt(n0)*(randn(1, Ns)+j*randn(1, Ns))/sqrt(2);
```

M2M4. m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          BLIND CRITERION           %
%       FOR AN ESPAR ANTENNA         %
%                MMMC                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function criterion=mmmc(y,M)

Num=mean(y.^M);
Den=mean(abs(y.^M).^2);
criterion=(abs(Num).^2)/(Den);
```

mmmc. m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%          BLIND CRITERION           %
%       FOR AN ESPAR ANTENNA         %
%         CRITERION: M2M4            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ky=m2m4(y,M)

Den=mean(abs(y).^2);
Num=mean(abs(y).^4);
Ky=-Num/(Den^2);
```

```
mod_CCK.m
%%%%%%%%%%%%%%%%%%%%%
%   CCK modulation  %
%%%%%%%%%%%%%%%%%%%%%
function [mod_data,phi1,nn] = mod_CCK(Ns,phi1,nn);
% N_data = 2^6;
data = [1 1 0 1 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1 0 0 0 1 0 0 1 0 1 0 0 0 0];
if nargin==1;
    data_block = data(1:8);

    phi0=0;
    phi2=bi2de([data_block(3) data_block(4)],'left-msb');
    phi3=bi2de([data_block(5) data_block(6)],'left-msb');
    phi4=bi2de([data_block(7) data_block(8)],'left-msb');

    a(1) = phi0+phi2+phi3+phi4;
    a(2) = phi0+phi3+phi4;
    a(3) = phi0+phi2+phi4;
    a(4) = phi0+phi4;
    a(5) = phi0+phi2+phi3;
    a(6) = phi0+phi3;
    a(7) = phi0+phi2;
    a(8) = phi0;

    mod_data0(1) = exp(j*pi/2*a(1));
    mod_data0(2) = exp(j*pi/2*a(2));
    mod_data0(3) = exp(j*pi/2*a(3));
    mod_data0(4) = -exp(j*pi/2*a(4));
    mod_data0(5) = exp(j*pi/2*a(5));
    mod_data0(6) = exp(j*pi/2*a(6));
    mod_data0(7) = -exp(j*pi/2*a(7));
    mod_data0(8) = exp(j*pi/2*a(8));

    phi1 = phi0;
    mod_data = mod_data0;
    for n = 2:ceil(Ns/8)
        nn = mod(n-1,4);
        const = 2;
        if nn == 2
            const = 0;
        end
        data_block = data(nn*8+1:nn*8+8);
        phi1=phi1+bi2de([data_block(1) mod(data_block(1)+data_block(2),2)],'left-msb')+const;
        phi2=bi2de([data_block(3) data_block(4)],'left-msb');
        phi3=bi2de([data_block(5) data_block(6)],'left-msb');
        phi4=bi2de([data_block(7) data_block(8)],'left-msb');

        a(1) = phi1+phi2+phi3+phi4;
        a(2) = phi1+phi3+phi4;
        a(3) = phi1+phi2+phi4;
```

```
        a(4)  = phi1+phi4;
        a(5)  = phi1+phi2+phi3;
        a(6)  = phi1+phi3;
        a(7)  = phi1+phi2;
        a(8)  = phi1;

        mod_data0(1)  = exp(j*pi/2*a(1));
        mod_data0(2)  = exp(j*pi/2*a(2));
        mod_data0(3)  = exp(j*pi/2*a(3));
        mod_data0(4)  = -exp(j*pi/2*a(4));
        mod_data0(5)  = exp(j*pi/2*a(5));
        mod_data0(6)  = exp(j*pi/2*a(6));
        mod_data0(7)  = -exp(j*pi/2*a(7));
        mod_data0(8)  = exp(j*pi/2*a(8));

        mod_data = [mod_data mod_data0];
    end
else
    mod_data=[];
    for n = 1:ceil(Ns/8)
        nn = mod(nn+1,4);
        const = 2;
        if nn == 2
            const = 0;
        end
        data_block = data(nn*8+1:nn*8+8);
        phi1=phi1+bi2de([data_block(1) mod(data_block(1)+data_block(2),2)],'left-msb')+const;
        phi2=bi2de([data_block(3) data_block(4)],'left-msb');
        phi3=bi2de([data_block(5) data_block(6)],'left-msb');
        phi4=bi2de([data_block(7) data_block(8)],'left-msb');

        a(1)  = phi1+phi2+phi3+phi4;
        a(2)  = phi1+phi3+phi4;
        a(3)  = phi1+phi2+phi4;
        a(4)  = phi1+phi4;
        a(5)  = phi1+phi2+phi3;
        a(6)  = phi1+phi3;
        a(7)  = phi1+phi2;
        a(8)  = phi1;

        mod_data0(1)  = exp(j*pi/2*a(1));
        mod_data0(2)  = exp(j*pi/2*a(2));
        mod_data0(3)  = exp(j*pi/2*a(3));
        mod_data0(4)  = -exp(j*pi/2*a(4));
        mod_data0(5)  = exp(j*pi/2*a(5));
        mod_data0(6)  = exp(j*pi/2*a(6));
        mod_data0(7)  = -exp(j*pi/2*a(7));
        mod_data0(8)  = exp(j*pi/2*a(8));
        mod_data = [mod_data mod_data0];
    end
```

```
end
```

SGA_Best criterion.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   BLIND ADAPTIVE BEAM FORMING   %
%       FOR AN ESPAR ANTENNA      %
%   STEEPEST GRADIENT ALGORITHM   %
%           M2M4 Vs MMMC          %
%         FINAL COMPARAISON       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all           % new workspace
clf                 % new figures

m=6;                % number of parasitic elements
z_s=50;             % output impedance of transmitter
Ns=64;              % number of samples
M=4;                % MPSK
N=50;                           % Iteration Number
SNR_dB=10;          % Initial SNR (in dB)
n0=10^(-SNR_dB/10); % noise power
delta_x=32;             % perturbation size
int_type=2;         % Signal type (1:QPSK, 2:RAND-PHASE PSK, 3:NO INTERFERENCE)
step_s=1500;                % step size
rep_sig=exp(-j*([0:M-1]/M*2*pi));
vol_size = [2048 -2047];
X_size = -(0.0217*vol_size+49.21);


% admittance matrix components: 6 components for a 7 element ESPAR antenna
y00=0.0008616-j*0.0120795;      % y00 parameter
y10=-0.0006963+j*0.0036462;     % y10 parameter
y11=0.0044216-j*0.0071600;      % y11 parameter
y21=0.0009721+j*0.0047851;      % y21 parameter
y31=-0.0005376-j*0.0011297;     % y31 parameter
y41=0.0001701-j*0.0002950;      % y41 parameter

% Y : admittance matrix
Y=[y00 y10 y10 y10 y10 y10 y10;
   y10 y11 y21 y31 y41 y31 y21;
   y10 y21 y11 y21 y31 y41 y31;
   y10 y31 y21 y11 y21 y31 y41;
   y10 y41 y31 y21 y11 y21 y31;
   y10 y31 y41 y31 y21 y11 y21;
   y10 y21 y31 y41 y31 y21 y11];

% inverse of the admittance matrix : Z
Z=inv(Y);

% DoA: Direction of Arrival=> 1st number: desired signal, others: interferences signals
DoA=[0 45];
```

```
% Initialisation
range = 300;
CDF = zeros(1, range);          %CDF
CDF_SINR = zeros(1, range);

ga =[];

J_all=[];
J_all_m2m4=[];
J_all_mmmc=[];
SINR=zeros(1, N+1);
es_SINR=zeros(1, N+1);
SINR_m2m4=zeros(1, N+1);
es_SINR_m2m4=zeros(1, N+1);
SINR_mmmc=zeros(1, N+1);
es_SINR_mmmc=zeros(1, N+1);


sample = 100;   % number of samples used for the simulation
ret=[];
ret_OF=[];
ret_CC=[];

Tablo=zeros(1, N);
Tablo_SINR=zeros(1, N+1);
Tab_Proba_m2m4=zeros(1, N);
Tab_Proba_mmmc=zeros(1, N);
Tab_BNR=zeros(1, sample);
 resolution=360;
 Tablo_Gain=zeros(1, resolution);
% Time display
start_p = time_disp1(sample);

% beginning of the sample loop
for ii = 1:sample

    % X : Reactance Matrix
    vol = zeros(1, m+1);
    X = diag(-(0.0217*vol+49.21)*j);
    X(1, 1)=z_s;

    % Unit Vector
    u_0=zeros(m+1, 1);
    u_0(1)=1;

    phi=DoA/360*2*pi;              % DoA [deg->rad]
    n_arrival=length(phi);
    power_dB = [3];
    power_i=10^(-power_dB/10);
    power=[1 power_i];
```

```
% steering vector
a_phi=zeros(m+1,n_arrival);
for i=1 :n_arrival
    a_phi(:,i)=[1 exp(j*pi/2*cos(phi(i)-[0:m-1]/m*2*pi))]';
end


% Vectors Initialization
J_tr=zeros(1,N);
JTr_m2m4=zeros(1,N);
JTr_mmmc=zeros(1,N);
dJ_dx=zeros(1,m+1);
dJ_dx_m2m4=zeros(1,m+1);         % DJ_dx for m2m4
dJ_dx_mmmc=zeros(1,m+1);         % DJ_dx for mmmc
dJ_dx_const_m2m4=zeros(1,m+1);         % DJ_dx for m2m4
dJ_dx_const_mmmc=zeros(1,m+1);         % DJ_dx for mmmc
SER=zeros(1,N);
sinr=zeros(1,N);
sinrMax = 0;                     % recording the max value of different sinr
Tab_J_m2m4=zeros(1,N);           % Table of gradient vector for m2m4 criterion
Tab_J_mmmc=zeros(1,N);           % Table of gradient vector for mmmc criterion
Tab_J_CriterionDiversity=zeros(1,N); % Table of gradient vector for criterion diversity
Tab_Norm_m2m4=zeros(1,N);        % Table of Norm(J) for m2m4
Tab_Norm_mmmc=zeros(1,N);        % Table of norm(J) for mmmc
Tab_Criterion=zeros(1,N);        % Recording the value of the criterion
grad_J=zeros(1,N);
grad_J_m2m4=zeros(1,N);
grad_J_mmmc=zeros(1,N);

gain_dBMin=0;                    % recording the value of the gain
criterion=1;                     % =1 => M2M4 ; =0 => MMMC


for n=1 :N      %SGA algorithm loop
    % CCK Modulation
    if n==1
        [sk,phi1,nn]=mod_CCK(Ns);
    else
        [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    end

    % Signal + interference + gaussian noise
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

     %  Choice of criterion: Determination of the two values of J_pre
    J_pre_m2m4=m2m4(y,M);
    J_pre_mmmc=mmmc(y,M);

    if criterion==1
```

```
    J_pre=J_pre_m2m4;
else
    J_pre=J_pre_mmmc;
end
%Tab_J_m2m4(n)=J_pre_m2m4; % M2M4 criterion
%Tab_J_mmmc(n)=J_pre_mmmc; % MMMC criterion

% w : Weight vector
w=2*X(1,1)*inv(Z+X)*u_0;
a=conj(w')*ref;

resolution=360;
phi_d=linspace(0,2*pi*(1-1/resolution),resolution);
ar=ones(m+1,resolution);

for i=2 :m+1
    ar(i,:)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
end

arout=abs(w'*ar);

gain_dB = 20*log10(arout/max(arout));
gain=(arout/max(arout));

Tab_BNR(n) = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);    % BNR
sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);           % SINR
Tablo(n)=Tablo(n)+(1/sample)*sinr(n);
  if (sinrMax<sinr(n))
    sinrMax = sinr(n);
end

X_test=X;

% for every parasitic element
for i=2:m+1
    X_test(i,i)=X_test(i,i)+j*delta_x;
    if imag(X(i,i)) > X_size(2)
        X(i,i) = X_size(2)*j;
    end
    if imag(X(i,i))<X_size(1)
        X(i,i)=X_size(1)*j;
    end

    % Signal modulation + interference + noise
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X_test,n0,power,Ns,M,m,int_type);

    % Determination of the two values of J
    J_m2m4=m2m4(y,M);
    J_mmmc=mmmc(y,M);
```

```
        dJ_dx_m2m4(i)=(J_m2m4-J_pre_m2m4)/delta_x;  % m2m4 gradient vector
        dJ_dx_mmmc(i)=(J_mmmc-J_pre_mmmc)/delta_x;  % mmmc gradient vector
        X_test(i,i)=X(i,i); % end of perturbation
    end

    Norm_dJ_dx_m2m4=norm(dJ_dx_m2m4);
    Norm_dJ_dx_mmmc=norm(dJ_dx_mmmc);

    % Value determination of the criterion
    if (Norm_dJ_dx_m2m4>Norm_dJ_dx_mmmc)
        criterion=1;          % m2m4 criterion
        Tab_Proba_m2m4(n)=Tab_Proba_m2m4(n)+1;
        dJ_dx=dJ_dx_m2m4;
    else
        criterion=0;          % mmmc criterion
        Tab_Proba_mmmc(n)=Tab_Proba_mmmc(n)+1;
        dJ_dx=dJ_dx_mmmc;
    end

    X=X+diag(j*step_s*dJ_dx);% update of the reactance matrix

    for i=2:m+1     % calibration of the reactance matrix
        if imag(X(i,i)) > X_size(2)
            X(i,i) = X_size(2)*j;
        end
        if imag(X(i,i))<X_size(1)
            X(i,i)=X_size(1)*j;
        end
    end
    % computing criterion norms
    Norm_dJ_dx_const_m2m4=norm(dJ_dx_const_m2m4);
    Norm_dJ_dx_const_mmmc=norm(dJ_dx_const_mmmc);
    % Recording of the values of Norm(m2m4) and Norm(mmmc)
    Tab_Norm_m2m4(n)=Norm_dJ_dx_m2m4;
    Tab_Norm_mmmc(n)=Norm_dJ_dx_mmmc;


    if criterion==1
        J_tr(n)=J_pre_m2m4;
        grad_J(n)=Norm_dJ_dx_m2m4;
    else
        J_tr(n)=J_pre_mmmc;
        grad_J(n)=Norm_dJ_dx_mmmc;
    end
    Tab_Criterion(n)=criterion; % updating the table
end % end of the SGA loop

resolution=360;
phi_d=linspace(0,2*pi*(1-1/resolution),resolution);
ar=ones(m+1,resolution);
```

```
    for i=2 :m+1
        ar(i,:)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
    end

    w=2*X(1,1)*inv(Z+X)*u_0; % weight vector
    % Signal modulation + interference + noise
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    % computing 2 criterions values
    J_pre_m2m4=m2m4(y,M);
    J_pre_mmmc=mmmc(y,M);

    if criterion==1
        J_pre=J_pre_m2m4; % m2m4 criterion choosed
    else
        J_pre=J_pre_mmmc;
    end

    J_tr(n+1)=J_pre;
    a=conj(w')*ref;

    sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2);      % SINR
    Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
    if (sinrMax<sinr(n))
        sinrMax = sinr(n);
    end

    arout=abs(w'*ar);
    gain_dB = 20*log10(arout/max(arout));
    gain=(arout/max(arout));
    for (y=1 : 360)
        if (gain_dB(y) < gain_dBMin)
            gain_dBMin=gain_dB(y);
        end
    end

    Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii; % beamforming gain
    BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1); % BNR
    Tab_BNR(ii)=BNR;
    ret = [ret ; BNR];
    cnt = round(BNR);
    cnt_s = round(sinr(end));
    SINR=(SINR*(ii-1)+sinr)/ii;
end
time_disp3(start_p);
```

SGA_m2m4.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  BLIND ADAPTIVE BEAM FORMING  %
%     FOR AN ESPAR ANTENNA      %
% (STEEPEST-GRADIENT ALGORITHM) %
%      CRITERION: M2M4          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all            % new workspace

m=6;                 % number of parasitic elements of an ESPAR antenna
z_s=50;              % output impedance
Ns=64;               % number of ESPAR ouptut sample used to compute criterion value
M=4;                 % MPSK
N=400;                       % number of iterations of the SGA algorithm
SNR_dB=10;           % Initial SNR (in dB)
n0=10^(-SNR_dB/10); % SNR
delta_x=32;                      % Delta_x parameter value
int_type=2;          % Type of interference (1:QPSK, 2:RAND-PHASE PSK, 3:NO INTERFERENCE)
step_s=1500;             % step size parameter value
rep_sig=exp(-j*([0:M-1]/M*2*pi));
vol_size = [2048 -2047];
X_size = -(0.0217*vol_size+49.21);

% admittance matrix components: 6 components for a 7 element ESPAR antenna
y00=0.0008616-j*0.0120795;  % y00 parameter
y10=-0.0006963+j*0.0036462; % y10 parameter
y11=0.0044216-j*0.0071600;  % y11 parameter
y21=0.0009721+j*0.0047851;  % y21 parameter
y31=-0.0005376-j*0.0011297; % y31 parameter
y41=0.0001701-j*0.0002950;  % y41 parameter

% admittance matrix Y
Y=[y00 y10 y10 y10 y10 y10 y10;
    y10 y11 y21 y31 y41 y31 y21;
    y10 y21 y11 y21 y31 y41 y31;
    y10 y31 y21 y11 y21 y31 y41;
    y10 y41 y31 y21 y11 y21 y31;
    y10 y31 y41 y31 y21 y11 y21;
    y10 y21 y31 y41 y31 y21 y11];

    % inverse of the admittance matrix : Z
    Z=inv(Y);
    % DoA: Direction of Arrival=> 1st number: desired signal, others: interferences signals
    DoA=[0 45];

    range = 90;
    CDF = zeros(1,range); %CDF
    CDF_SINR = zeros(1,range);
    ga =[];
    J_all=[];
```

```
%initialisation
SINR=zeros(1,N+1);
es_SINR_dB=zeros(1,N+1);
Tablo_SINR=zeros(1,N+1);
resolution=360;
xax=linspace(0,2*pi,resolution);
Tablo_Gain=zeros(1,resolution);
Tab_J=zeros(1,N);

sample = 1000; % number of samples used for the simulation
ret=[];
Tab_BNR=zeros(1,sample);
start_p = time_disp1(sample);

% beginning of the sample loop
for ii = 1:sample
    % reactance matrix
    vol = zeros(1,m+1);
    X = diag(-(0.0217*vol+49.21)*j);
    X(1,1)=z_s;

    % unit vector
    u_0=zeros(m+1,1);
    u_0(1)=1;


    phi=DoA/360*2*pi;                   % DoA [deg->rad]
    n_arrival=length(phi);
    power_dB=[0];
    power_i=10^(-power_dB/10);
    power=[1 power_i];
    a_phi=zeros(m+1,n_arrival);
    % steering vector
    for i=1:n_arrival
        a_phi(:,i)=[1 exp(j*pi/2*cos(phi(i)-[0:m-1]/m*2*pi))]';
    end
    % initialisation
    J_tr=zeros(1,N);
    dJ_dx=zeros(1,m+1);
    SER=zeros(1,N);
    sinr=zeros(1,N);
    grad_J=zeros(1,N);
    %main loop
    for n=1:N
        % CCK modulation
        if n==1
            [sk,phi1,nn]=mod_CCK(Ns);
        else
            [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        end
```

```
    [ref, y]=esper_out(sk, a_phi, Z, X, n0, power, Ns, M, m, int_type); % ESPAR signal modelisation

    % criterion value: M2M4
    J_pre=m2m4(y, M);
    Tab_J(n)=J_pre;
    w=2*X(1, 1)*inv(Z+X)*u_0; % weight vector
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);   % SINR

    X_test=X;

    % for every parasitic element
    for i=2 :m+1
        % update of the reactance matrix
        X_test(i, i)=X_test(i, i)+j*delta_x;
        if imag(X(i, i)) > X_size(2)
            X(i, i) = X_size(2)*j;
        end
        if imag(X(i, i))<X_size(1)
            X(i, i)=X_size(1)*j;
        end
        % signal modelisation
        [sk, phi1, nn]=mod_CCK(Ns, phi1, nn);
        [ref, y]=esper_out(sk, a_phi, Z, X_test, n0, power, Ns, M, m, int_type);
        J=m2m4(y, M);     % criterion value: M2M4

        dJ_dx(i)=(J-J_pre)/delta_x; % gradient vector
        X_test(i, i)=X(i, i);
    end
    % update of the reactance matrix
    X=X+diag(j*step_s*dJ_dx);
    % calibration for every parasitic element
    for i=2:m+1
        if imag(X(i, i)) > X_size(2)
            X(i, i) = X_size(2)*j;
        end
        if imag(X(i, i))<X_size(1)
            X(i, i)=X_size(1)*j;
        end
    end
    J_tr(n)=J_pre;
    grad_J(n)=norm(dJ_dx);
end


resolution=360;
phi_d=linspace(0, 2*pi*(1-1/resolution), resolution);
ar=ones(m+1, resolution);
for i=2 :m+1
    ar(i, :)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
```

```
        end
        w=2*X(1,1)*inv(Z+X)*u_0; % weight vector
        % signal modelisation
        [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);
        % criterion value : M2M4
        J_pre=m2m4(y,M);
        J_tr(n+1)=J_pre;
        a=conj(w')*ref;
        sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2);% SINR

        Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
        arout=abs(w'*ar);

        % CDF
        gain_dB = 20*log10(arout/max(arout));
        Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii;
        BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
        Tab_BNR(ii)  = BNR;
        ret = [ret ; BNR];
        cnt = round(BNR);
        SINR_dB = 10*log10(sinr(end));
        cnt_s = round(SINR_dB);
        SINR=(SINR*(ii-1)+sinr)/ii;
        for i=1 :length(J_tr)
            sinr=abs(J_tr(i))^2/(1-(abs(J_tr(i)))^2);
            es_SINR_dB0(i)=10*log10(sinr);
        end
        es_SINR_dB=(es_SINR_dB*(ii-1)+es_SINR_dB0)/ii;
        if cnt_s>-21
            for jcdf = -20:cnt_s
                CDF_SINR(1,jcdf+21)  = CDF_SINR(1,jcdf+21)+1;
            end
        end
        if cnt>-21
            for jcdf = -20:cnt
                CDF(1,jcdf+21)  = CDF(1,jcdf+21)+1;
            end
            ga((cnt+21),:)  = gain_dB;
            J_all((cnt+21),:)  = J_tr;

            if mod(ii,50)  == 0
                time_disp2(start_p,ii);
            end
        end
    end
end
time_disp3(start_p);
```

SRA_BestCriterion.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   BLIND ADAPTIVE BEAM FORMING    %
%       FOR AN ESPAR ANTENNA       %
%   SEQUENTIAL RANDOM ALGORITHM    %
%           M2M4 Vs MMMC           %
%         FINAL COMPARAISON        %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all       % new workspace
clf             % new figures
m=6;                % number of parasitic elements
z_s=50;             % output impedance of transmitter
Ns=64;              % number of samples
M=4;                % MPSK
N=50;                           % Iteration Number
SNR_dB=10;          % Initial SNR (in dB)
n0=10^(-SNR_dB/10); % noise power
int_type=2;         % Signal type (1:QPSK, 2:RAND-PHASE PSK, 3:NO INTERFERENCE)
rep_sig=exp(-j*([0:M-1]/M*2*pi));
vol_size = [-2048 2047];
SIGMA = [2000];     % sigma parameter

% admittance matrix components: 6 components for a 7 element ESPAR antenna

y00=0.0008616-j*0.0120795;  % y00 parameter
y10=-0.0006963+j*0.0036462; % y10 parameter
y11=0.0044216-j*0.0071600;  % y11 parameter
y21=0.0009721+j*0.0047851;  % y21 parameter
y31=-0.0005376-j*0.0011297; % y31 parameter
y41=0.0001701-j*0.0002950;  % y41 parameter

% Y : admittance matrix
Y=[y00 y10 y10 y10 y10 y10 y10;
    y10 y11 y21 y31 y41 y31 y21;
    y10 y21 y11 y21 y31 y41 y31;
    y10 y31 y21 y11 y21 y31 y41;
    y10 y41 y31 y21 y11 y21 y31;
    y10 y31 y41 y31 y21 y11 y21;
    y10 y21 y31 y41 y31 y21 y11];

    % inverse of the admittance matrix : Z
    Z=inv(Y);

    % DoA: Direction of Arrival=> 1st number: desired signal, others: interferences signals
    DoA=[0 45];

    % Initialisation
    range = 100;
    CDF = zeros(1,range);%CDF
    CDF_SINR = zeros(1,range);
```

```
ga = [];
J_all = [];
J_all2 = [];
es_SINR_dB=zeros(1,N);
resolution=360;
sample = 1000;      % number of samples used for the simulation
ret=[];
start_p = time_disp1(sample);
Tablo_Gain=zeros(1,resolution);
Tab_BNR=zeros(1,sample);
Tablo_SINR=zeros(1,N+1);


for ii = 1:sample    % beginning of the sample loop

    % reactance matrix
    vol = randint(m,1,vol_size);
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);

    % unit vector
    u_0=zeros(m+1,1);
    u_0(1)=1;

    % steering vector
    phi=DoA/360*2*pi;                  %DoA [deg->rad]
    n_arrival=length(phi);  % Number of arriving signals
    power_dB=[0];
    power_i=10^(-power_dB/10);
    power=[1 power_i];
    a_phi=zeros(m+1,n_arrival);
    for i=1 :n_arrival
        a_phi(:,i)=[1 exp(j*pi/2*cos(phi(i)-[0:m-1]/m*2*pi))]';
    end

    % initialisation
    J_tr=zeros(1,N);
    dJ_dx=zeros(1,m+1);
    SER=zeros(1,N);
    sinr=zeros(1,N);
    grad_J=zeros(1,N);

    % ESPAR signal modelisation
    [sk,phi1,nn]=mod_CCK(Ns);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    % computing criterion values
    J_pre_m2m4=m2m4(y,M);
    J_pre_mmmc=mmmc(y,M);
```

```
w=2*X(1,1)*inv(Z+X)*u_0;     %w: weight vector
a=conj(w')*ref;

sinr(1)=mean(abs(a).^2)/mean(abs(y-a).^2);% SINR

n = 1;

while n < N
    n = n+1;
    % Random preturbation
    sigma=round(SIGMA/(1+n/250));
    volshift=randn(m,1)*sigma;  % for guassian distrubution
    volold=vol;
    vol=vol+volshift;
    % calibration
    for index=1:m                  % -REAMax<=rea<=REAMax
        if vol(index) > vol_size(2)
            vol(index) = vol_size(2);
        elseif vol(index)<vol_size(1)
            vol(index)=vol_size(1);
        end
    end
    % random change in the reactance matrix
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);
    % ESPAR signal modelisation
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    % compute both criterions values
    J_m2m4=m2m4(y,M);
    J_mmmc=mmmc(y,M);

    w=2*X(1,1)*inv(Z+X)*u_0;     % weight vector
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);   % SINR

    % compute criteria's values differences
    Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
    Diff_mmmc=(J_mmmc)-(J_pre_mmmc);

    if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
        % m2m4 criterion choosed
        J_pre_m2m4 = J_m2m4;
        if n == N
            break
        end
    elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
        % mmmc criterion choosed
        J_pre_mmmc = J_mmmc;
```

```
    if n == N
        break
    end
else
    % none of the criteria increase the cost
    vol=volold;
    if n == N
        break
    end
    n = n+1;
    % another random change is made
    vol=vol-volshift;
    for index=1:m           % -REAMax<=rea<=REAMax
        if vol(index) > vol_size(2)
            vol(index) = vol_size(2);
        elseif vol(index)<vol_size(1)
            vol(index)=vol_size(1);
        end
    end
    X = -(0.0217*vol+49.21);
    X = diag([z_s;X*j]);
    % ESPAR signal modelisation
    [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
    [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);

    % compute criterion values
    J_m2m4=m2m4(y,M);
    J_mmmc=mmmc(y,M);


    w=2*X(1,1)*inv(Z+X)*u_0;     % weight vector
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);% SINR
    % compute criteria's values differences
    Diff_m2m4=(J_m2m4)-(J_pre_m2m4);
    Diff_mmmc=(J_mmmc)-(J_pre_mmmc);

    if (Diff_m2m4>Diff_mmmc) & (Diff_m2m4>=0)
        % m2m4 criterion choosed
        J_pre_m2m4 = J_m2m4;
        if n == N
            break
        end
    elseif (Diff_mmmc>Diff_m2m4) & (Diff_mmmc>=0)
        % mmmc criterion choosed
        J_pre_mmmc = J_mmmc;
        if n == N
            break
        end
    else
```

```
            vol=volold;
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %X = -(0.0217*vol+49.21);
            %X = diag([z_s;X*j]);
            %[sk, phi1, nn]=mod_CCK(Ns, phi1, nn);
            %[ref, y]=esper_out(sk, a_phi, Z, X, n0, power, Ns, M, m, int_type);

            %w=2*X(1, 1)*inv(Z+X)*u_0;
            %a=conj(w')*ref;
            %sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            if n == N
                break
            end
        end
    end
end

% steering vector
resolution=360;
phi_d=linspace(0, 2*pi*(1-1/resolution), resolution);
ar=ones(m+1, resolution);
for i=2 :m+1
    ar(i, :)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
end
% update of the ractance matrix
X = -(0.0217*vol+49.21);
X = diag([50;X*j]);
w=2*X(1, 1)*inv(Z+X)*u_0;
arout=abs(w'*ar);


sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2); % SINR
Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
% CDF
gain_dB = 20*log10(arout/max(arout));
Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii;
BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
Tab_BNR(ii)=BNR;


ret = [ret ; BNR];
cnt = round(BNR);
SINR_dB = 10*log10(sinr(end));
cnt_s = round(SINR_dB);

if cnt_s>-20
    for jcdf = -20:cnt_s
        CDF_SINR(1, jcdf+21) = CDF_SINR(1, jcdf+21)+1;
    end
```

```
            if cnt > -20
                for jcdf = -20:cnt
                    CDF(1, jcdf+21) = CDF(1, jcdf+21)+1;
                end

                ga((cnt+21), :) = gain_dB;

            end

            if mod(ii,50) == 0
                time_disp2(start_p, ii);
                fprintf('DoA = [%3i %3i]¥n', [DoA])
            end
        end
    end

end

time_disp3(start_p);
```

```
SRA_m2m4.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BLIND ADAPTIVE BEAM FORMING   %
%      FOR AN ESPAR ANTENNA      %
% (STEEPEST-GRADIENT ALGORITHM) %
%       CRITERION: M2M4          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all             % new workspace

m=6;                  % number of parasitic elements of an ESPAR antenna
z_s=50;               % output impedance
Ns=64;                % number of ESPAR ouptut sample used to compute criterion value
M=4;                  % MPSK
N=400;                        % number of iterations of the SGA algorithm
SNR_dB=10;            % Initial SNR (in dB)
n0=10^(-SNR_dB/10); % SNR
delta_x=32;                   % Delta_x parameter value
int_type=2;           % Type of interference (1:QPSK, 2:RAND-PHASE PSK, 3:NO INTERFERENCE)
step_s=1500;              % step size parameter value
rep_sig=exp(-j*([0:M-1]/M*2*pi));
vol_size = [2048 -2047];
X_size = -(0.0217*vol_size+49.21);

% admittance matrix components: 6 components for a 7 element ESPAR antenna
y00=0.0008616-j*0.0120795;  % y00 parameter
y10=-0.0006963+j*0.0036462; % y10 parameter
y11=0.0044216-j*0.0071600;  % y11 parameter
y21=0.0009721+j*0.0047851;  % y21 parameter
y31=-0.0005376-j*0.0011297; % y31 parameter
y41=0.0001701-j*0.0002950;  % y41 parameter

% admittance matrix Y
Y=[y00 y10 y10 y10 y10 y10 y10;
    y10 y11 y21 y31 y41 y31 y21;
    y10 y21 y11 y21 y31 y41 y31;
    y10 y31 y21 y11 y21 y31 y41;
    y10 y41 y31 y21 y11 y21 y31;
    y10 y31 y41 y31 y21 y11 y21;
    y10 y21 y31 y41 y31 y21 y11];

    % inverse of the admittance matrix : Z
    Z=inv(Y);
    % DoA: Direction of Arrival=> 1st number: desired signal, others: interferences signals
    DoA=[0 45];

    range = 90;
    CDF = zeros(1,range); %CDF
    CDF_SINR = zeros(1,range);
    ga =[];
    J_all=[];
```

```
%initialisation
SINR=zeros(1,N+1);
es_SINR_dB=zeros(1,N+1);
Tablo_SINR=zeros(1,N+1);
resolution=360;
xax=linspace(0,2*pi,resolution);
Tablo_Gain=zeros(1,resolution);
Tab_J=zeros(1,N);

sample = 1000; % number of samples used for the simulation
ret=[];
Tab_BNR=zeros(1,sample);
start_p = time_disp1(sample);

% beginning of the sample loop
for ii = 1:sample
    % reactance matrix
    vol = zeros(1,m+1);
    X = diag(-(0.0217*vol+49.21)*j);
    X(1,1)=z_s;

    % unit vector
    u_0=zeros(m+1,1);
    u_0(1)=1;


    phi=DoA/360*2*pi;                    % DoA [deg->rad]
    n_arrival=length(phi);
    power_dB=[0];
    power_i=10^(-power_dB/10);
    power=[1 power_i];
    a_phi=zeros(m+1,n_arrival);
    % steering vector
    for i=1 :n_arrival
        a_phi(:,i)=[1 exp(j*pi/2*cos(phi(i)-[0:m-1]/m*2*pi))]';
    end
    % initialisation
    J_tr=zeros(1,N);
    dJ_dx=zeros(1,m+1);
    SER=zeros(1,N);
    sinr=zeros(1,N);
    grad_J=zeros(1,N);
    %main loop
    for n=1 :N
        % CCK modulation
        if n==1
            [sk,phi1,nn]=mod_CCK(Ns);
        else
            [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        end
```

```
    [ref, y]=esper_out(sk, a_phi, Z, X, n0, power, Ns, M, m, int_type); % ESPAR signal modelisation

    % criterion value: M2M4
    J_pre=m2m4(y, M);
    Tab_J(n)=J_pre;
    w=2*X(1, 1)*inv(Z+X)*u_0; % weight vector
    a=conj(w')*ref;
    sinr(n)=mean(abs(a).^2)/mean(abs(y-a).^2);   % SINR

    X_test=X;

    % for every parasitic element
    for i=2 :m+1
        % update of the reactance matrix
        X_test(i, i)=X_test(i, i)+j*delta_x;
        if imag(X(i, i)) > X_size(2)
            X(i, i) = X_size(2)*j;
        end
        if imag(X(i, i))<X_size(1)
            X(i, i)=X_size(1)*j;
        end
        % signal modelisation
        [sk, phi1, nn]=mod_CCK(Ns, phi1, nn);
        [ref, y]=esper_out(sk, a_phi, Z, X_test, n0, power, Ns, M, m, int_type);
        J=m2m4(y, M);     % criterion value: M2M4

        dJ_dx(i)=(J-J_pre)/delta_x; % gradient vector
        X_test(i, i)=X(i, i);
    end
    % update of the reactance matrix
    X=X+diag(j*step_s*dJ_dx);
    % calibration for every parasitic element
    for i=2:m+1
        if imag(X(i, i)) > X_size(2)
            X(i, i) = X_size(2)*j;
        end
        if imag(X(i, i))<X_size(1)
            X(i, i)=X_size(1)*j;
        end
    end
    J_tr(n)=J_pre;
    grad_J(n)=norm(dJ_dx);
end


resolution=360;
phi_d=linspace(0, 2*pi*(1-1/resolution), resolution);
ar=ones(m+1, resolution);
for i=2 :m+1
    ar(i, :)=exp(j*pi/2*cos(phi_d-(i-2)*2*pi/m));
```

```
        end
        w=2*X(1,1)*inv(Z+X)*u_0; % weight vector
        % signal modelisation
        [sk,phi1,nn]=mod_CCK(Ns,phi1,nn);
        [ref,y]=esper_out(sk,a_phi,Z,X,n0,power,Ns,M,m,int_type);
        % criterion value : M2M4
        J_pre=m2m4(y,M);
        J_tr(n+1)=J_pre;
        a=conj(w')*ref;
        sinr(n+1)=mean(abs(a).^2)/mean(abs(y-a).^2);% SINR

        Tablo_SINR=(Tablo_SINR*(ii-1)+sinr)/ii;
        arout=abs(w'*ar);

        % CDF
        gain_dB = 20*log10(arout/max(arout));
        Tablo_Gain=(Tablo_Gain*(ii-1)+gain_dB)/ii;
        BNR = gain_dB(DoA(1)+1)-gain_dB(DoA(2)+1);
        Tab_BNR(ii) = BNR;
        ret = [ret ; BNR];
        cnt = round(BNR);
        SINR_dB = 10*log10(sinr(end));
        cnt_s = round(SINR_dB);
        SINR=(SINR*(ii-1)+sinr)/ii;
        for i=1 :length(J_tr)
            sinr=abs(J_tr(i))^2/(1-(abs(J_tr(i)))^2);
            es_SINR_dB0(i)=10*log10(sinr);
        end
        es_SINR_dB=(es_SINR_dB*(ii-1)+es_SINR_dB0)/ii;
        if cnt_s>-21
            for jcdf = -20:cnt_s
                CDF_SINR(1,jcdf+21) = CDF_SINR(1,jcdf+21)+1;
            end
        end
        if cnt>-21
            for jcdf = -20:cnt
                CDF(1,jcdf+21) = CDF(1,jcdf+21)+1;
            end
            ga((cnt+21),:) = gain_dB;
            J_all((cnt+21),:) = J_tr;

            if mod(ii,50) == 0
                time_disp2(start_p,ii);
            end
        end
    end
end
time_disp3(start_p);
```

time_disp3.m

```
%%%%%%%%%%%%%%%%%%%%%%%
%       Time_disp 3     %
%        function       %
%%%%%%%%%%%%%%%%%%%%%%%
function time_disp3(start_p)

co_sec = [24*3600 3600 60 1];
finish_t=clock;

fprintf('¥ndate %d j %d h %d mn', [finish_t(3:5)])

all_sec=sum((finish_t(3:6)-start_p(3:6)).*co_sec);

day=fix(all_sec/(24*3600));                 % days
hour=mod(fix(all_sec/3600), 24);            % hours
min=mod(fix(all_sec/60), 60);               % minutes
sec=fix(mod(all_sec, 60));                  % seconds

fprintf('¥n computing time : %d days  %d hours  %d minutes  %d seconds ?@', day, hour, min, sec)
```