

TR-ACR-0004

003

Interference Cancellation in OFDM Systems and  
Reactance-Domain MUSIC Algorithm for ESPAR Antennas

Cyril Plapous, Jun Cheng, Eddy Taillefer,  
Masaya Hashiguchi, Akifumi Hirata and Takashi Ohira

2002.9.30

(株)国際電気通信基礎技術研究所  
適応コミュニケーション研究所

〒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  
Adaptive Communications Research Laboratories  
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan  
Telephone: +81-774-95-1501 Fax: +81-774-95-1508

©2001 (株)国際電気通信基礎技術研究所

©2001 Advanced Telecommunications Research Institute International



## Abstracts and Key words

On the one hand, this report describes an attempt to apply the Electronically Steerable Parasitic Array Radiator (ESPAR) antenna to an Orthogonal Frequency Division Multiplexing (OFDM) scheme to cancel the delayed signals out of the guard interval. Computer simulation results show good performance for the case where signals impinging on the ESPAR antenna are both in and out of the guard interval. Thanks to its simple hardware structure, the ESPAR antenna can be used aboard user terminals to improve their capacity. On the other hand, a new technique of estimation for Direction-of-Arrivals (DoAs) is proposed for the ESPAR antenna. This technique is based on the modified MUSIC (Multiple Signal Classification) algorithm. An  $M$ -dimension vector  $\mathbf{y}$  of the output of the antenna is obtained by making a circular shift of a given vector of the reactances  $M$  times. Then the  $M$  by  $M$  correlation matrix of the  $\mathbf{y}$  vector is employed to estimate the MUSIC spectrum. Simulation results show that DoAs can be estimated by the reactance-domain MUSIC algorithm for the ESPAR antenna.

**Key words:** ESPAR antenna, multipath cancellation, adaptive beamforming, gradient-based algorithm, MCCC, OFDM, reactance-domain MUSIC algorithm, direction-of-arrival, direction finder, mobile user terminals, low complexity.



# Acknowledgements

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

I also want to thank Mr. Akifumi Hirata for helping me after the departure of my supervisor.

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. Atsuya Ando, Ms. Qing Han, Mr. Masaya Hashiguchi, Mr. Akifumi Hirata, Mr. Kyoichi Iigusa, Mr. Keizo Inagaki, Mr. Hiroki Tanaka, Ms. Tomoko Tanaka, Mr. Satoru Tawara and Mr. Kehu Yang, for their collaboration and discussions and for having very good parties 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 their fantastic help to prepare all the procedures to come in Japan and for their kind personality. I hope they can speak french very well now !

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



# Contents

Abstracts and Key words	3
Acknowledgements	5
Glossary	15
Notations	17
Introduction	19
<b>1 ATR Adaptive Communications Research Laboratories</b>	<b>21</b>
1.1 Introduction to ATR . . . . .	21
1.2 ACR Laboratories . . . . .	22
1.2.1 Organisation . . . . .	22
1.2.2 Research domains . . . . .	23
1.2.3 Main purpose project . . . . .	23
<b>2 Adaptive ESPAR antenna</b>	<b>25</b>
2.1 Wireless Ad-Hoc Community Network - Concept and features	25
2.1.1 Routing schemes . . . . .	26
2.1.2 Space Division Multiple Access (SDMA) . . . . .	26
2.1.3 User terminals . . . . .	26
2.1.4 Microwave signal processing . . . . .	26
2.2 ESPAR antenna and signal model . . . . .	29
2.2.1 ESPAR antenna configuration . . . . .	29
2.2.2 ESPAR antenna model . . . . .	30
2.2.3 Signal model . . . . .	32
<b>3 ESPAR Antenna in an OFDM scheme</b>	<b>35</b>
3.1 Introduction . . . . .	35
3.2 Orthogonal Frequency Division Multiplexing . . . . .	35
3.3 ESPAR antenna in an OFDM scheme . . . . .	38
3.4 Computer simulations with MATLAB . . . . .	42
3.4.1 Efficacy of ESPAR antenna in an OFDM scheme . . . . .	42

3.4.2	Cancellation of long time delayed paths . . . . .	43
3.4.3	Performances in term of BER . . . . .	45
3.5	Conclusion . . . . .	46
<b>4</b>	<b>Reactance-domain MUSIC algorithm for ESPAR antenna</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Correlation matrix of a single-port antenna . . . . .	48
4.3	Reactance-domain MUSIC algorithm . . . . .	50
4.4	Computer simulations with MATLAB . . . . .	54
4.5	Discussions for practical cases . . . . .	62
4.6	Conclusion . . . . .	62
	<b>Conclusion</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>
	<b>Appendix A : MatLab source codes for ESPAR + OFDM</b>	<b>69</b>
	OFDM.m . . . . .	69
	pilot.m . . . . .	78
	steer_vector.m . . . . .	79
	deg2rad.m . . . . .	80
	reactances2current.m . . . . .	81
	receiver.m . . . . .	82
	calculation_BER.m . . . . .	83
	polar_ploth.m . . . . .	84
	<b>Appendix B : MatLab source codes for AAA + MUSIC</b>	<b>87</b>
	create_matrix_aaa_line.m . . . . .	87
	music_aaa_line.m . . . . .	89
	<b>Appendix C : MatLab source codes for AAA with mutual coupling + MUSIC</b>	<b>91</b>
	create_matrix_aaa_round_mc.m . . . . .	91
	music_aaa_round_mc.m . . . . .	94
	steer_vector_roundmc.m . . . . .	96
	degree2radian.m . . . . .	97
	<b>Appendix D : MatLab source codes for ESPAR + MUSIC</b>	<b>99</b>
	create_matrix_espar.m . . . . .	99
	music_espar.m . . . . .	102
	create_matrix_espar_multipath.m . . . . .	104
	music_espar_multipath.m . . . . .	107
	create_matrix_espar_no_periodic.m . . . . .	109
	music_espar_no_periodic.m . . . . .	112
	steer_vector_esparnew.m . . . . .	114



degree2radian.m . . . . .	115
reactances2currentbis.m . . . . .	116



# List of Figures

2.1	DBF - Digital Beamforming . . . . .	27
2.2	RBF - RF Beamforming . . . . .	27
2.3	ABF - Aerial Beamforming . . . . .	28
2.4	Configuration of an ESPAR antenna . . . . .	29
2.5	Geometry of an ESPAR antenna . . . . .	32
3.1	OFDM symbol in frequency domain . . . . .	36
3.2	Guard Interval insertion . . . . .	37
3.3	ESPAR antenna in OFDM scheme . . . . .	38
3.4	Flowchart of the adaptive algorithm . . . . .	41
3.5	BER for ESPAR antenna (solid curve) and omni-directional antenna (dashed curve) . . . . .	42
3.6	ESPAR antenna pattern . . . . .	43
3.7	Depressed signals before decision . . . . .	44
3.8	ESPAR antenna pattern . . . . .	45
3.9	Depressed signals before decision . . . . .	45
3.10	BER versus the number of delayed paths . . . . .	46
4.1	Correlation matrix for conventional array antenna . . . . .	49
4.2	Correlation matrix in reactance domain . . . . .	50
4.3	Flowchart of the reactance-domain MUSIC algorithm . . . . .	52
4.4	MUSIC spectrum for random set of reactances, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal . . . . .	54
4.5	MUSIC spectrum for random set of reactances, DoAs=170°, 180°, SNR=5 dB for each signal . . . . .	55
4.6	Circular antenna with mutual coupling . . . . .	55
4.7	MUSIC spectrum comparison between conventional array an- tenna and ESPAR antenna, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal . . . . .	56
4.8	MUSIC spectrum comparison between shifted method and random method for the set of reactance, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal . . . . .	56
4.9	MUSIC spectrum for SNR=20, 5 and 0 dB, DoAs=60°, 200°, 240°, 310° . . . . .	57

4.10	MUSIC spectrum for periodic SOI (DoA=90°) and 3 no periodic interferences (DoAs=150°, 210° and 300°) . . . . .	58
4.11	MUSIC spectrum in multipath environment, DoAs=50°, 120°, 230°, 320° . . . . .	59
4.12	MUSIC spectrum for narrow range of reactances, DoAs=60°, 200°, 240°, 310° . . . . .	60
4.13	MUSIC spectrum for N=1000 and N=50, DoAs=40°, 130°, 210°, 300° . . . . .	61

## List of Tables

3.1	First simulation assumptions . . . . .	42
3.2	Second simulation assumptions . . . . .	43
3.3	Third simulation assumptions . . . . .	44
4.1	DoA estimation for different SNR . . . . .	57
4.2	DoA estimation for the signal of interest . . . . .	58
4.3	DoA estimation in multipath environment . . . . .	60
4.4	DoA estimation for large and narrow range of reactance . . . . .	60
4.5	Influence of the number of samples on the DoA estimation . . . . .	61



# Glossary

- ABF:** Aerial BeamForming
- BER:** Binary Error Rate
- DBF:** Digital BeamForming
- DoA:** Direction of Arrival
- ESPAR:** Electronically Steerable Parasitic Array Radiator
- FFT:** Fast Fourier Transform
- GI:** Guard Interval
- ICI:** Inter-Carrier Interference
- IEEE:** Institute of Electrical and Electronics Engineers
- IEICE:** Institute of Electronics, Information and Communication Engineers
- IFFT:** Inverse Fast Fourier Transform
- ISI:** Inter-Symbol Interference
- ISM:** Industrial, Scientific and Medical
- LMS:** Least Mean Square
- MBF:** Microwave BeamForming
- MCCC:** Maximum Cross Correlation Coefficient
- MCM:** MultiCarrier Modulation
- MMSE:** Minimum Mean-Square Error
- MUSIC:** Multiple Signal Classification
- OFDM:** Orthogonal Frequency Division Multiplexing
- QPSK:** Quadrature Phase Shift Keying

**RCS:** Radio Communications Society

**RF:** Radio Frequency

**SDMA:** Space Division Multiple Access

**SOI:** Signal Of Interest

**SNR:** Signal to Noise Ratio

**WACNet :** Wireless Ad-hoc Community Network

**WCNC:** Wireless Communications and Networking Conference



# Notations

In this part we want to settle the notations used in this report for all equations and calculus.

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  $\mathbf{E}[\ ]$  operator realizes the stochastic mean of a process.

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



# Introduction

With the development of multimedia and mobile communications, the variety of networks and terminals is increasing and communication environments are becoming more complex. Adaptive Communications Research (ACR) laboratories are conducting research on fundamental technologies for flexible mobile multimedia communication systems which autonomously adapt to changes in environments so users are not aware of any restrictions on communications.

Recently, the concept of the Wireless Ad-hoc Community Network (WACNet) has shown much promise. Such a network can be considered as a means of linking portable user terminals that meet temporarily in locations where connection to a network infrastructure is difficult.

The application of advanced adaptive techniques has contributed to the development of high-performance receiving antennas with the capability of automatically eliminating surrounding interferences. Conventionally, it has been thought that only digital technology could provide these advantages. ATR has proposed Electronically Steerable Parasitic Array Radiator (ESPAR) antennas as a means of reviving analog adaptive beamforming, which provides a dramatically simplified architecture that results in significantly lower power dissipation and fabrication cost than are common with digital beamforming antennas.

In this report, there are two distinct purposes, the first part of our work was to find a way to apply the ESPAR antenna to an Orthogonal Frequency Division Multiplexing (OFDM) scheme and the second part of our work was to find a solution to apply the conventional Multiple Signals Classification (MUSIC) algorithm to the ESPAR antenna.

After presenting the WACNet concept, the ESPAR antenna features will be explained. Then, we will present and explain our work to apply the ESPAR antenna to an OFDM scheme and comment the simulation results. Next, we will describe the new concept developed for the ESPAR antenna in order to create the reactance-domain MUSIC algorithm for the ESPAR antenna and show our simulation results. These two works will be published and presented in conferences of Institute of Electronics, Information and Communication Engineers (IEICE) and Institute of Electrical and Electronics Engineers (IEEE).



## 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 four following actual departments:

- Adaptive Communications Research Laboratories (ACR),
- Human Information Science Laboratories (HIS),
- Media Information Science Laboratories (MIS),
- Spoken Language Translation Research Laboratories (SLT).

## 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.

ACR is researching segmentation algorithms for broader use of wireless resources, routing, media access control, and adaptive antennas with the aim of realizing "WACNet", an autonomous distributed network which does not need infrastructure.

"Networking Anywhere" is technology components for autonomous distributed networks, aimed at supporting free communication by data transmission in ad-hoc gatherings of people.

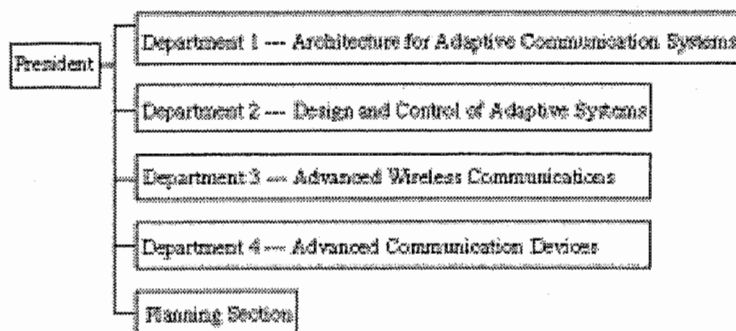
### 1.2.1 Organisation

**Established on March 27, 1996**

**Research Period : March 1996 to February 2003**

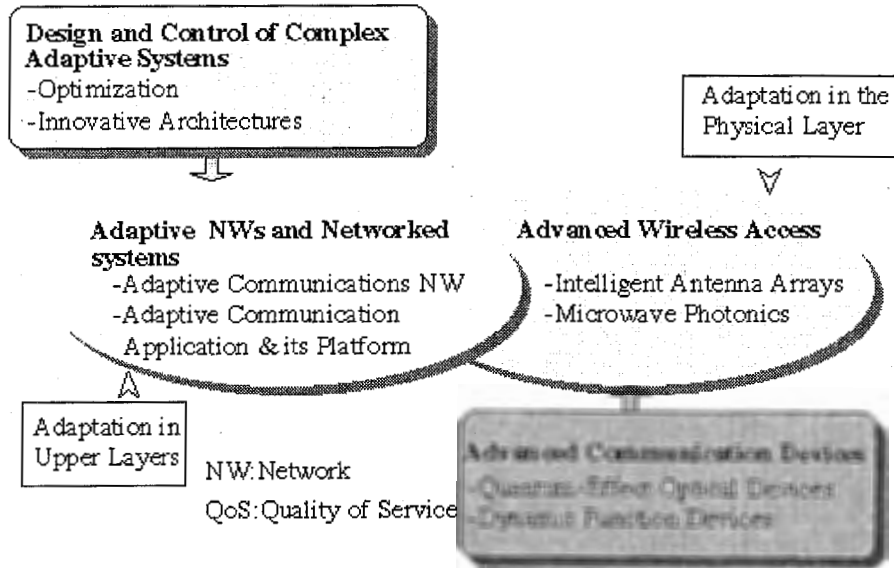
**Total Budget for 7 years : ¥ 11,900 million**

#### Organization

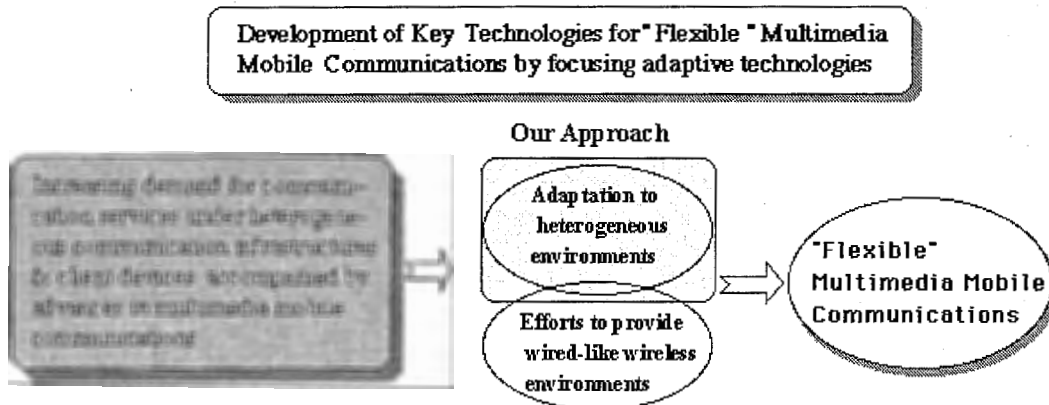


### 1.2.2 Research domains

#### -Adaptive Technologies for Multimedia Wireless Communications-



### 1.2.3 Main purpose project



Typical NW variation		Physical variation among clients				Data rate in IMT-2000	
NW	Bandwidth	Terminal	Memory	Screen size	Bits/pixel	Mobility	Date rate (Ws)
LAN	10-100M	PC	64MB	1024x768	16b/p	Fixed	2M
ISDN	128k	laptop	16MB	800x600	8b/p	Movable	384k
PHS	32k	PDA	2MB	320x200	2b/p	Fast	144k





## Chapter 2

# Adaptive ESPAR antenna

In this chapter, the WACNet concept will be presented, in order to situate the ESPAR antenna in this new concept. Next, the ESPAR antenna configuration and modelisation will be detailed. A signal model will also be proposed for the ESPAR antenna.

### 2.1 Wireless Ad-Hoc Community Network - Concept and features

Recently, the concept of Wireless Ad-Hoc Community Networks (WACNet) [1] has become a hot topic. Such network can be considered as a means of communications among portable user terminals that temporarily meet, where distance and time come close yet easy connection to a network infrastructure is not possible. For example, Bluetooth [2] has been developed to connect between mobile handy phones, headsets, PCs, and other devices using the ISM (Industrial, Scientific and Medical) band (2.4 GHz). Bluetooth has great possibilities to enable wireless ad-hoc networks. The concept of WACNet is proposed in order to have more scalability, more speeds, and low output power level, that makes it very attractive for mobile network use.

The WACNet consists of portable user terminals only and has no infrastructure such as hosts, base stations, switches, and hubs. It is therefore expected to be applicable anywhere; not only in fixed space such as a office company but also in a moving space such as a bus or train. It can also be used in a disaster stricken area where the existing telecommunications infrastructure has been damaged. The WACNet has scalability, *i.e.*, the ability to accept a large number of users terminals because of autonomous segmentation and routing. The key technologies developed to achieve the WACNet are routing schemes, Space Division Multiple Access (SDMA), user terminals, microwave signal processing, and Electronically Steerable Parasitic Array Radiator (ESPAR) antennas.

### 2.1.1 Routing schemes

WACNet routing is carried out by each user terminal autonomously. This means that each user terminal has switching and relay functions and routes are made by the user terminals through the use of these functions voluntarily. In these routing schemes, dynamic topologies such as the number of terminals and the relative locations and velocities among terminals must be considered.

### 2.1.2 Space Division Multiple Access (SDMA)

To achieve scalability, the adoption of dynamic network segmentation is considered. With dynamic network segmentation, a network is divided into a number of segments consisting of some terminals. As one channel is able to be used repeatedly in the different segments, the SDMA technique is very useful since it enables the limited network resources to be effectively utilized. Packets from one terminal to terminals in the same segment are delivered directly and packets to terminals in different segments are delivered via relay terminals.

### 2.1.3 User terminals

Since the WACNet is infrastructure free, the only costs involved depend on the user terminals. To make the WACNet popular it is necessary to miniaturize and reduce the cost and weight of the user terminals. One of the important technologies for terminal weight reduction is to reduce the battery weight because the battery occupies a significant percentage of the weight of the user terminal. Accordingly, if an adaptive antenna can be implemented in the user terminal, a large amount of battery power can be saved because the adaptive antenna has a higher gain than an omni-directional antenna, and can therefore effectively reduce the RF (Radio Frequency) power necessary to transmit packets. The adaptive antenna can also suppress co-channel interference and can therefore enlarge the capacity of a network, *i.e.*, the number of terminals that can be accommodated in the same network.

### 2.1.4 Microwave signal processing

Adaptive antennas essentially have a great advantage in saving RF power. However, the high cost and high power consumption in their signal processing units still prevent their use from spreading to the general population in uses such as user terminals. The present adaptive antennas are Digital Beamforming (DBF) antennas (see Figure 2.1), which need the same

number of RF high-power or low-noise amplifiers, frequency converters, and D/A or A/D converters as the elements in conventional array antennas.

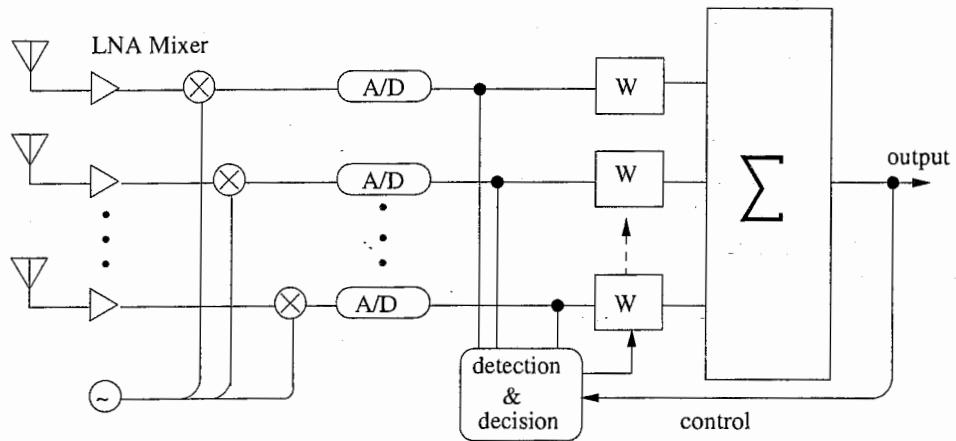


Figure 2.1: DBF - Digital Beamforming

Therefore, the implementation of adaptive antennas in user terminals had been said to be impractical. This is now possible with the "Microwave signal processing" concept using RF Beamforming (RBF) (see Figure 2.2) instead of DBF.

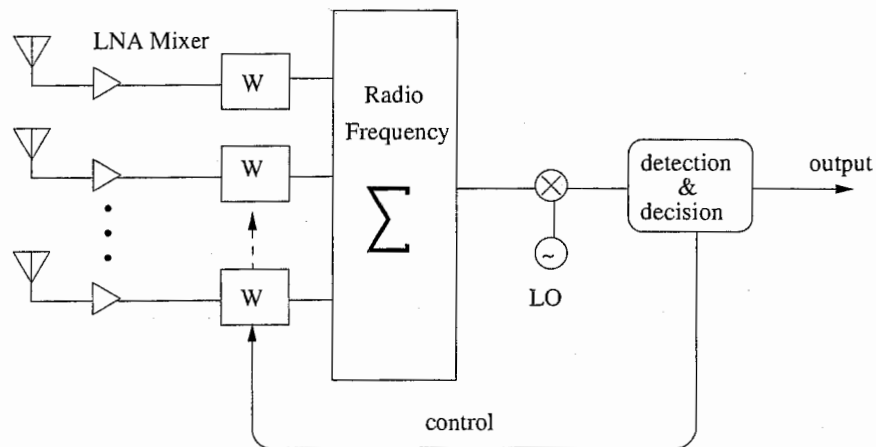


Figure 2.2: RBF - RF Beamforming

This new beamforming architecture has been proposed to achieve the ultimate reductions in the size, weight, power dissipation, and fabrication cost of adaptive antennas. The operating principle of this architecture is

based upon electromagnetic coupling among array elements. It is called "Aerial Beamforming" (ABF) (see Figure 2.3) because signal combining is carried out in space, not in circuits. Since this array requires only one RF port to feed, the RF circuit scale is drastically reduced compared with other configurations.

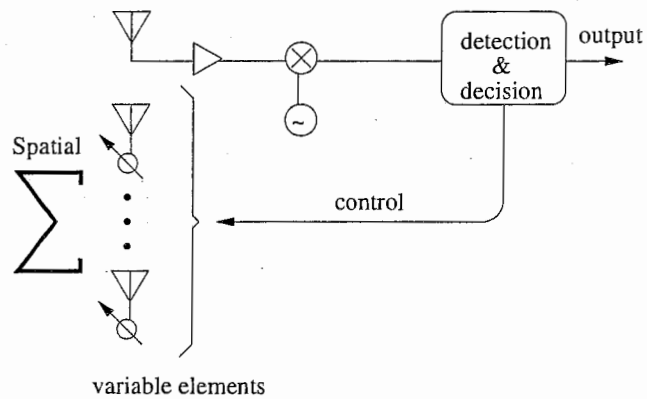


Figure 2.3: ABF - Aerial Beamforming

## 2.2 ESPAR antenna and signal model

As a practical hardware realization of the ABF concept, the Electronically Steerable Parasitic Array Radiator (ESPAR) antenna [3, 4] has been proposed. In this section, we review the ESPAR antenna configuration and detail its model. Next we give the signal model used in our simulations for the ESPAR antenna [3].

### 2.2.1 ESPAR antenna configuration

The ESPAR antenna consists of  $(M + 1)$  elements with  $M = 6$  in our configuration, which is depicted in Figure 2.4.

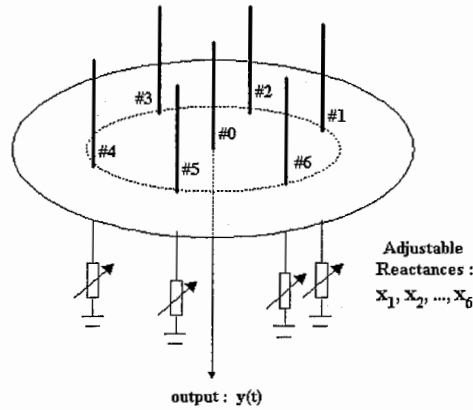


Figure 2.4: Configuration of an ESPAR antenna

This antenna is a reactively controlled array with only one active radiator. The 0-th element is the active radiator located at the center of a circular ground plane. The  $M$  other elements are parasitic radiators symmetrically surrounding the active radiator. The pattern of this antenna is controlled by adjusting the values  $x_m$  ( $m = 1, 2, \dots, M$ ) of the reactances connected to the parasitic radiators. Thus, the ESPAR antenna can work in two modes transmit-mode and receive-mode. In practical applications, the reactances may be constrained in certain ranges, *e.g.*, from  $-300 \Omega$  to  $300 \Omega$ . The vector denoted by

$$\mathbf{x} = [x_1, x_2, \dots, x_m, \dots, x_M]^T \quad (2.1)$$

is called the reactance vector and is used to form the ESPAR antenna pattern.

We now briefly formulate the output of the ESPAR antenna. Let's denote  $\mathbf{s}(t) = [s_0(t), s_1(t), \dots, s_m(t), \dots, s_M(t)]^T$ , where the component  $s_m(t)$  is the RF signal impinging on the  $m$ -th element of the ESPAR antenna. Then the single-port output  $y(t)$  of the antenna is given by

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

where  $\mathbf{i} = [i_0, i_1, \dots, i_m, \dots, i_M]^T$  is the RF current vector with the component  $i_m$  appearing on the  $m$ -th element.

### 2.2.2 ESPAR antenna model

Now we will derive the RF current vector  $\mathbf{i}$  as a function of the reactance vector. For convenience, we derive the model of the ESPAR antenna in a transmit mode. The theorem of reciprocity tells us that the received-mode radiation pattern of an antenna is equal to that of the transmit-mode. Therefore all the following results are also available in the receive-mode.

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

$$v_0 = V_s - Z_0 i_0, \quad (2.3)$$

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

$$v_m = -j x_m i_m, \quad m = 1, 2, \dots, M. \quad (2.4)$$

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

$$\mathbf{v} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \\ \vdots \\ v_M \end{bmatrix} = \begin{bmatrix} V_s - Z_0 i_0 \\ -j x_1 i_1 \\ \vdots \\ -j x_m i_m \\ \vdots \\ -j x_M i_M \end{bmatrix} = V_s \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \mathbf{X} \mathbf{i}, \quad (2.5)$$

where  $\mathbf{X} = \text{diag}[Z_0, j x_1, \dots, j x_M]$  is called the reactance matrix. In addition, the RF current vector  $\mathbf{i}$  and RF current voltage  $\mathbf{v}$  have the relationship

$$\mathbf{i} = \mathbf{Y} \mathbf{v}, \quad (2.6)$$

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

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

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 :

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

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).

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

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

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

$$\begin{aligned} y_{11} &= y_{22} = y_{33} = y_{44} = y_{55} = y_{66} \\ y_{01} &= y_{02} = y_{03} = y_{04} = y_{05} = y_{06} \\ y_{12} &= y_{23} = y_{34} = y_{45} = y_{56} = y_{61} \\ y_{13} &= y_{24} = y_{35} = y_{46} = y_{51} = y_{62} \\ y_{14} &= y_{25} = y_{36} \end{aligned} \quad (2.10)$$

Finally, equations (2.9) and (2.10) imply that the admittance matrix  $\mathbf{Y}$  is determined by only 6 components of the mutual admittances [3]  $y_{00}$ ,  $y_{10}$ ,  $y_{11}$ ,  $y_{21}$ ,  $y_{31}$  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

$$\mathbf{Y} = \begin{bmatrix} y_{00} & y_{10} & & & & & y_{10} \\ y_{10} & y_{11} & y_{21} & y_{31} & y_{41} & y_{31} & y_{21} \\ & y_{21} & \ddots & \ddots & \ddots & \ddots & y_{31} \\ \vdots & y_{31} & \ddots & \ddots & \ddots & \ddots & y_{41} \\ & y_{41} & \ddots & \ddots & \ddots & \ddots & y_{31} \\ & y_{31} & \ddots & \ddots & \ddots & \ddots & y_{21} \\ y_{10} & y_{21} & y_{31} & y_{41} & y_{31} & y_{21} & y_{11} \end{bmatrix}. \quad (2.11)$$

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

The  $\mathbf{i}$  current vector can also be formulated as [3]

$$\mathbf{i} = V_s(\mathbf{Z} + \mathbf{X})^{-1}\mathbf{u}_0, \quad (2.12)$$

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

It should be emphasized that the signal vector  $\mathbf{s}(t)$  impinging on the elements of the ESPAR antenna is not measurable. This differs from the conventional adaptive array antennas where the received signal vector on the elements is observed. For the ESPAR antenna, only the single-port output  $y(t)$  can be measured. More unfortunately, the single-port output  $y(t)$  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.

### 2.2.3 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 2.5.

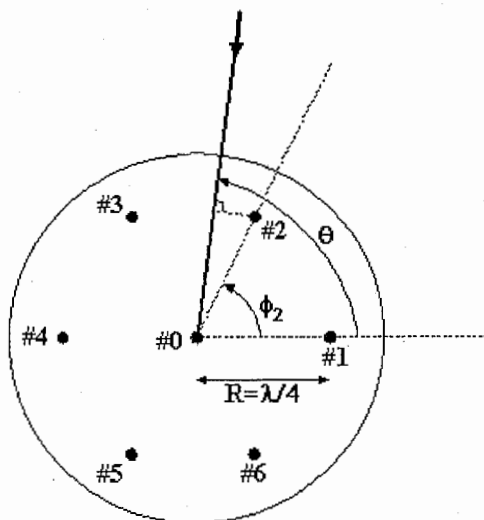


Figure 2.5: Geometry of an ESPAR antenna

The  $m$ -th element is placed at an angle  $\phi_m = \frac{2\pi}{M}(m - 1)$ , ( $m =$



$1, 2, \dots, M$ ) relative to an arbitrary axis. When an incoming wavefront is impinging on the antenna from a Direction-of-Arrival (DoA) of  $\theta$  relative to the same reference axis, there is a spatial delay of  $R\cos(\theta - \phi_m)$  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  $\lambda$ , this spatial delay can be expressed as an electrical angular difference defined by  $\frac{2\pi}{M}R\cos(\theta - \phi_m)$ . For the ESPAR antenna the radius is  $R = \frac{\lambda}{4}$  then the steering vector for a DoA of  $\theta$  can be expressed as [3]

$$\mathbf{a}(\theta) = \begin{bmatrix} 1 \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_1)} \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_2)} \\ \vdots \\ e^{j\frac{\pi}{2}\cos(\theta-\phi_M)} \end{bmatrix}. \quad (2.13)$$

Suppose there are a total number of  $D$  transmitted signals  $u_d(t)$  with direction-of-arrivals (DoAs)  $\theta_d$  ( $d = 1, \dots, D$ ). Let  $s_m(t)$  ( $m = 0, 1, \dots, M$ ) 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

$$\mathbf{s}(t) = \sum_{d=1}^D \mathbf{a}(\theta_d)u_d(t), \quad (2.14)$$

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

According to this signal model, we can give the output of the ESPAR antenna as [3]

$$\mathbf{y}(t) = \mathbf{i}^T \mathbf{s}(t) + n(t) = \sum_{d=1}^D \mathbf{i}^T \mathbf{a}(\theta_d)u_d(t) + n(t), \quad (2.15)$$

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  $\mathbf{i}(t) = [i_0(t), \dots, i_m(t), \dots, i_M(t)]^T$ , with the component  $i_m(t)$  appearing on the  $m$ -th element. Notice that we will voluntarily omit the  $t$  from the current vector  $\mathbf{i}(t)$  for the sake of simplicity and finally, the notation for the current vector will be  $\mathbf{i}$ .

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.



## Chapter 3

# ESPAR Antenna in an OFDM scheme

### 3.1 Introduction

Orthogonal Frequency Division Multiplexing (OFDM) [6, 7] achieves a fast transmission rate over multipath fading channels. This is a wideband modulation method that divides the available bandwidth into multiple subchannels to profit from the benefits possessed by a wideband signal under multipath fading. The orthogonality of the carriers allows their individual spectra to overlap which provides desirable properties, such as multipath and echo immunity. But if a path has a delay greater than the guard interval (GI), the performance is drastically degraded.

Adaptive antennas [8] can be used in conjunction with OFDM to increase system capacity by using spatial diversity to compensate for channel impairments. However, adaptive antennas are generally implemented in base stations because of their high complexity.

The Electronically Steerable Parasitic Array Radiator (ESPAR) antenna [3] has only a single-port output, and has shown high potential for applications to wireless terminal devices. Since all the signals on the antenna element cannot be observed, conventional adaptive array processing cannot be applied to the ESPAR antenna. In this chapter, we attempt to apply the ESPAR antenna to an OFDM scheme to improve the performance of user terminals.

### 3.2 Orthogonal Frequency Division Multiplexing

OFDM is in class of modulation techniques known as Multicarrier Modulation (MCM). This is a wideband modulation method that divides the

available bandwidth into multiple subchannels to profit of the benefits possessed by a wideband signal under multipath fading.

In a normal frequency division multiplexing (FDM) system the carriers are separated by a guard band to allow them to be received and demodulated through conventional filtering procedures. These guard bands result in a lower spectral efficiency. For OFDM, the carriers are mathematically orthogonal so they can be arranged so that their sidebands overlap while still allowing reception without interference from adjacent carriers or Inter-Carrier Interference (ICI). This OFDM characteristic (see Figure 3.1) provides a good efficiency in bandwidth usage and so fast data rates transmissions with some desirable properties like multipath and echo immunity.

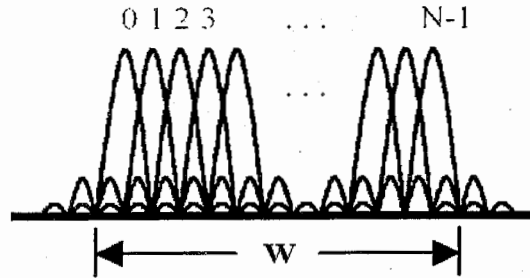


Figure 3.1: OFDM symbol in frequency domain

Each carrier  $\Phi_n(t)$  for each subchannel can be thought of as sinusoids or exponentials of the form

$$\Phi_n(t) = \begin{cases} \frac{1}{\sqrt{T}} \exp(j2\pi \frac{W}{N} kt) & \text{if } t \in [0, T] \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

spaced  $W/N$  Hz apart, where  $W$  is the available bandwidth. Each carrier is scaled by a complex constellation value  $x_{n,m}$  from the input data; the subscript  $n$  corresponds to the index number of the carrier, and  $m$  is the index of the entire OFDM symbol. For a continuous transmission,  $m$  is an integer,  $m \in ]-\infty, \infty[$ . The constellation points most commonly come from  $M$ -ary Quadrature Phase Shift Keying (QPSK) or  $M$ -ary Quadrature Amplitude Modulation ( $M$ -QAM), but almost any mapping will work. The scaled carriers are then summed to yield the time waveform to be transmitted over the channel,

$$s_m(t) = \sum_{n=0}^{N-1} x_{n,m} \Phi_n(t - mT). \quad (3.2)$$

An infinite sequence of OFDM symbols or frames is a juxtaposition of all the individual OFDM symbols, and we drop the index  $m$  to give

$$s(t) = \sum_{m=-\infty}^{\infty} s_m(t) = \sum_{m=-\infty}^{\infty} \sum_{n=0}^{N-1} x_{n,m} \Phi_n(t - mT). \quad (3.3)$$

Since  $\Phi_n(t)$  is a rectangular pulse modulated on the carrier frequency  $kW/N$ , OFDM is commonly viewed as having  $N$  subcarriers, each carrying the lower symbol rate  $R_{OFDM} = R_S/N$ , where  $R_S$  is the symbol rate of the input data. Note the symbol rate of each subchannel is the rate at which OFDM symbols or frames are sent.

The OFDM receiver is a bank of demodulators, so if carriers other than the desired carrier all mixed down to frequencies that have a whole number of cycles in the symbol period ( $\tau$ ), then they will integrate to zero over the symbol period. Thus, the carriers will be linearly independent, or orthogonal, if the carrier spacing is a multiple of  $1/\tau$ . Mathematically, a set of functions will be orthogonal if

$$\int_a^b \Psi_p(t) \Psi_q^*(t) dt = \begin{cases} K & \text{for } p = q \\ 0 & \text{for } p \neq q \end{cases}. \quad (3.4)$$

The most famous set of orthogonal function are the complex exponentials which form the basis of the Fourier transform:

$$\Psi_k(t) = e^{j\omega_k t}, \quad \omega_k = \omega_0 + 2\pi \frac{k}{t}. \quad (3.5)$$

Their orthogonality makes them good candidates for OFDM transmission and suggests the Direct Fourier Transform (DFT) as a valuable method for generating efficiently the transmission waveform knowing that the DFT can be calculated easily and cheaply using the Fast Fourier Transform (FFT).

One way to prevent Inter-Symbol Interference (ISI) is to create a cyclically extended guard interval (GI), like it is shown in Figure 3.2. If the GI is longer than the multipath delays, the ISI is eliminated and the remaining (ICI) can easily be suppressed by an one-tap equalizer for each subchannel in frequency domain. The tap coefficient are calculated based on channel information. The inverse fast Fourier transform (IFFT) is an easy and efficient way to perform the OFDM modulator while demodulator is performed simply by a FFT.

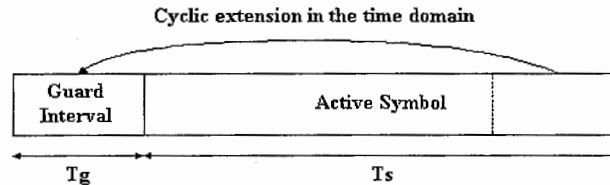


Figure 3.2: Guard Interval insertion

### 3.3 ESPAR antenna in an OFDM scheme

Our purpose is to cancel the delayed paths out of the GI using the ESPAR antenna while the OFDM demodulator suppresses the effects of delayed paths in the GI.

Since the ESPAR antenna has only one output port, the proposed adaptive antenna array requires only one FFT (Fast Fourier Transform) device. So it is like the pre-FFT equalization [9] for a conventional array antenna, which requires only one FFT device, in contrast to the post-FFT which requires the same number of FFTs as the number of sensors used for the antenna and so demands a high complexity for the receiver.

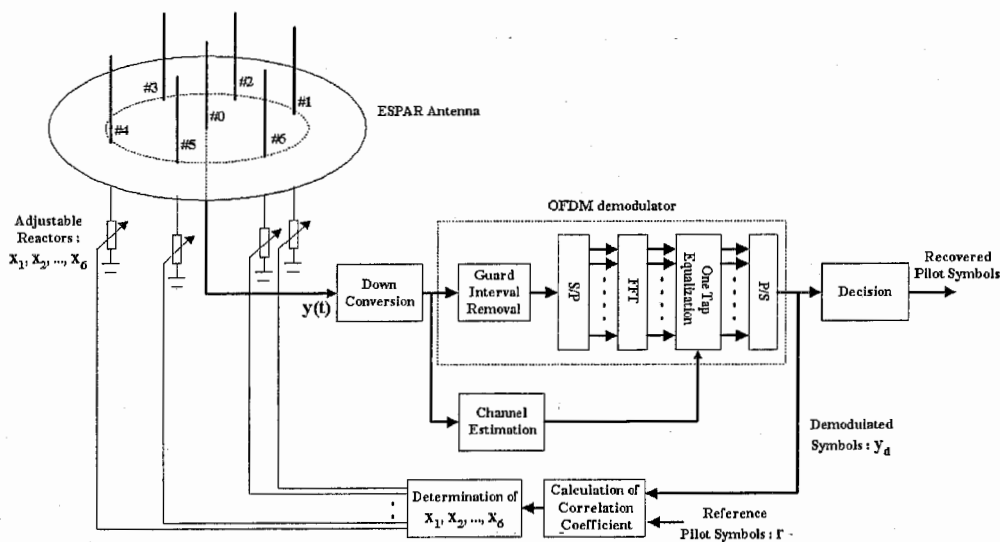


Figure 3.3: ESPAR antenna in OFDM scheme

In the receiver (see Figure 3.3), the ESPAR antenna receives the impinging signals and its single-port output is down-converted and demodulated by the OFDM demodulator. To achieve the OFDM demodulation, we first remove the GI, next we compute the FFT of the OFDM symbol, and finally we equalize each subchannel in the frequency domain. The one-tap equalizer coefficients are found by channel estimation, assumed to be perfect, and the number of channel samples estimated is the same as the size of the GI. In practice, the channel estimation can be implemented by a temporal FIR (Finite Impulse Response) filter using the LMS (Least Mean Square) algorithm to adapt and find the temporal coefficients of the channel. For the equalization we compute the FFT of the estimated channel to obtain the coefficients used to make the correction of the subchannels.

Thus, the OFDM demodulator will only suppress the effects of the delayed path in the GI and we will use the demodulated signal  $y_d$  (see Fig-

ure 3.3) to make the ESPAR antenna cancel the delayed path out of the GI, which strongly degrades the system performance. This way we can preserve the degree of freedom of the ESPAR antenna by avoiding the creation of some useless nulls.

The gradient-based algorithm [3, 5], based on the MCCC (Maximum Cross Correlation Coefficient) criterion, is used to adapt the pattern of the ESPAR antenna. Using the demodulated signal  $\mathbf{y}_d$  we calculate the correlation coefficient [3] as follows

$$\rho_n = \frac{|\mathbf{y}_d^H(n)\mathbf{r}(n)|}{\sqrt{\mathbf{y}_d^H(n)\mathbf{y}_d(n)}\sqrt{\mathbf{r}^H(n)\mathbf{r}(n)}}, \quad (3.6)$$

where  $\mathbf{r}(n)$  is a reference pilot sequence with a  $P$ -dimension like  $\mathbf{y}_d(n)$ . According to this coefficient, we determine the reactance values to update the antenna pattern.

A gradient vector of  $\rho_n$  is defined as:

$$\nabla_{\rho_n} = \frac{\partial \rho_n}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \rho_n}{\partial x_1} \\ \frac{\partial \rho_n}{\partial x_2} \\ \vdots \\ \frac{\partial \rho_n}{\partial x_M} \end{bmatrix}, \quad (3.7)$$

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

The aim is to maximize the cross correlation coefficient. When it is close to unity, the output of the ESPAR antenna and our reference signal become similar and thus, we can consider that the convergence is reached and the nulls and the beam are formed.

For conventional adaptive array antenna, the objective function (MMSE) is quadratic (convex) with respect to the weight vector, thus requiring the linear filtering Wiener theory for the optimization problem. With the ESPAR antenna, the objective function  $\rho_n$  is a non-convex function with respect to the reactance vector  $\mathbf{x}$ . Thus, we must resort to the nonlinear filtering theory, which has not been thoroughly studied and applied to adaptive beamforming. This is the basic difficulty. Furthermore, in general, the error performance surface of the iterative procedure may have local minima in addition to a global minimum, and more than one global minimum may exist.

With the steepest gradient algorithm we proceed as follows:

- (i) Begin with an initial value  $\mathbf{x}$  of the reactance vector, which is chosen arbitrarily. Typically,  $\mathbf{x}$  is set equal to the zero vector, where the initial pattern is omni-directional.
- (ii) Using this initial or present guess, compute the gradient vector  $\nabla_{\rho_n}$  at time  $n$  (*i.e.*, at the  $n$ -th iteration).

- (iii) Calculate the next guess at the reactance vector by making a change in the initial or present guess in a direction similar to that of the gradient vector.
- (iv) Go back to step (ii) and repeat the process.

Let  $\mathbf{x}(n)$  denote the value of reactance vector  $\mathbf{x}$  at time  $n$ . According to the steepest gradient method described above, the update value of the reactance vector at time  $n + 1$  is computed by using the simple relation

$$\mathbf{x}(n + 1) = \mathbf{x}(n) + \mu \nabla \rho_n, \quad n = 1, 2, \dots, N \quad (3.8)$$

where  $\mu$  is a positive real-value constant that controls the convergence speed.

An estimate value of the gradient vector  $\nabla_{\rho_n}$  (see Equation 3.7) may be obtained by using finite-difference approximations of derivatives. Here, the first-order partial derivative  $\frac{\partial \rho_n}{\partial x_i}$  can be use. The first order partial derivate  $\frac{\partial \rho_n}{\partial x_i}$  with respect to the reactance  $x_i$  is approximated to a change of the cross correlation coefficient

$$\frac{\partial \rho_n}{\partial x_m} \approx \frac{\rho_n(x_1, x_2, \dots, x_m + \Delta x_m, \dots, x_M) - \rho_n(x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_m} \quad (3.9)$$

by incrementing  $x_m$  to  $x_m + \Delta x_m$ , where  $\Delta x_m$  corresponds to the perturbation size. By using these estimate values, the gradient vector can be written like this :

$$\nabla \rho_n \approx \begin{bmatrix} \frac{\rho_n(x_1 + \Delta x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_1} \\ \frac{\rho_n(x_1, x_2 + \Delta x_2, \dots, x_m, \dots, x_M)}{\Delta x_2} \\ \vdots \\ \frac{\rho_n(x_1, x_2, \dots, x_m + \Delta x_m, \dots, x_M)}{\Delta x_m} \\ \vdots \\ \frac{\rho_n(x_1, x_2, \dots, x_m, \dots, x_M + \Delta x_M)}{\Delta x_M} \end{bmatrix} - \begin{bmatrix} \frac{\rho_n(x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_1} \\ \frac{\rho_n(x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_2} \\ \vdots \\ \frac{\rho_n(x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_m} \\ \vdots \\ \frac{\rho_n(x_1, x_2, \dots, x_m, \dots, x_M)}{\Delta x_M} \end{bmatrix} \quad (3.10)$$

In practice, each component of gradient vector is considered having the same perturbation size, *i.e.*,  $\Delta x_1 = \Delta x_2 = \dots = \Delta x_M = \Delta x$ .

The reactances are sequentially perturbed to determine the gradient vector used to update the reactance vector. The gradient vector has an  $M$ -dimension, and only one component of this vector is calculated at a time from the output of the antenna, so we must repeat this step  $M$  times to obtain the complete gradient vector. That is to say we have to send  $M + 1$  times the same reference vector  $\mathbf{r}(n)$  for each iteration.

Because of the OFDM characteristic of working with a block of samples, we will set  $P$ , the dimension of the pilot sequence, to the size of the FFT.



This way we won't lose samples while making the adaptation of the ESPAR antenna pattern. So we need  $M + 1$  OFDM symbols to make one adaptation step for the ESPAR antenna. Thus the total  $P(M + 1)N$  samples are needed for  $N$  iterations, which is equivalent to  $(M + 1)N$  OFDM symbols. The flowchart of the proposed adaptive algorithm is illustrated in Figure 3.4.

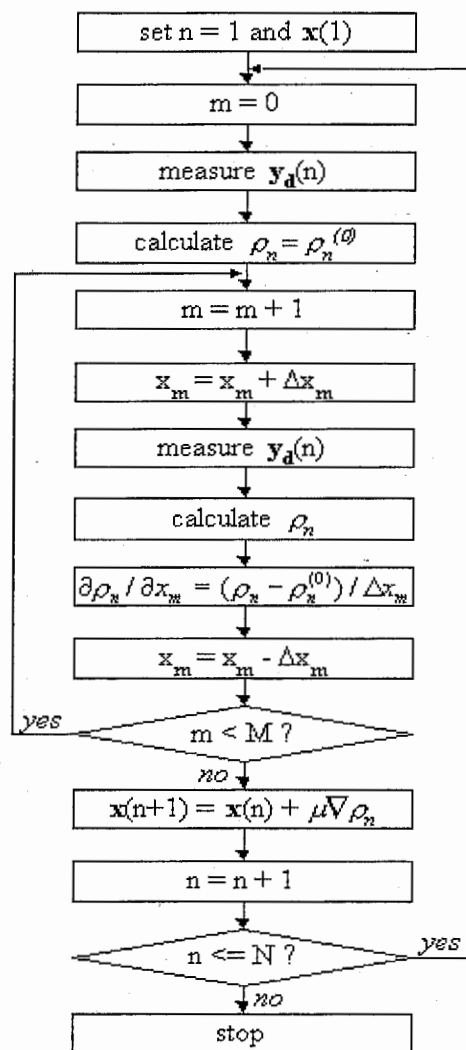


Figure 3.4: Flowchart of the adaptive algorithm

### 3.4 Computer simulations with MATLAB

The multipath signals are delayed copies of the main path as a result of reflections from fixed terrain features. Since the delayed copies have DoA different from that of the main path, we can use spatial diversity to cancel the multipath components. In our simulations, the FFT size is 64 points and the GI length is 16 samples. The step-size parameter  $\mu$  is set to 550, the perturbation size parameter  $\Delta x$  is set to 5 and we assume that the data symbols to recover are in QPSK modulation.

#### 3.4.1 Efficacy of ESPAR antenna in an OFDM scheme

For the first simulation, we assume only two paths with different DoAs, the main path and a delayed copy of this path out of the GI (see Table 3.1), so an OFDM demodulator with an omni-directional antenna is not sufficient to cancel the interferences induced by the delayed path because the path is out of the GI.

Table 3.1: First simulation assumptions

	Impinging signals	
	1	2
DoA	80°	210°
Delay (samples)	0	25
Power (normalized)	1	0.5

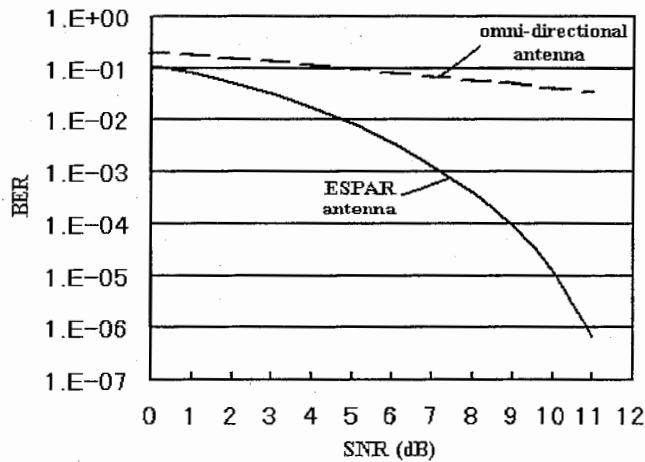


Figure 3.5: BER for ESPAR antenna (solid curve) and omni-directional antenna (dashed curve)

Figure 3.5 shows two Binary Error Rate (BER) curves. The dashed line is the BER using an omni-directional antenna with the OFDM demodulator and shows that the BER becomes poor when even only one delayed path is out of the GI. The solid line is the BER obtained by applying the adaptive ESPAR antenna to an OFDM scheme and shows that the enhancement provided by using the ESPAR antenna is important and necessary to reach good BER values.

### 3.4.2 Cancellation of long time delayed paths

For the second simulation, we assume four paths impinging on the ESPAR antenna: the main path and three delayed copies of this path (see Table 3.2). The path with 21 samples is clearly out of the GI, and cannot be canceled in an OFDM receiver with an omni-directional antenna. By using the ESPAR antenna in an OFDM scheme (see Figure 3.3), this path signal is efficiently canceled (see Figure 3.6) whereas the 2 other paths are not canceled by the ESPAR antenna to preserve the degree of freedom of the antenna.

Table 3.2: Second simulation assumptions

	Impinging signals			
	1	2	3	4
DoA	0°	60°	280°	180°
Delay (samples)	0	5	12	21
Power (normalized)	1	0.5	0.5	0.5

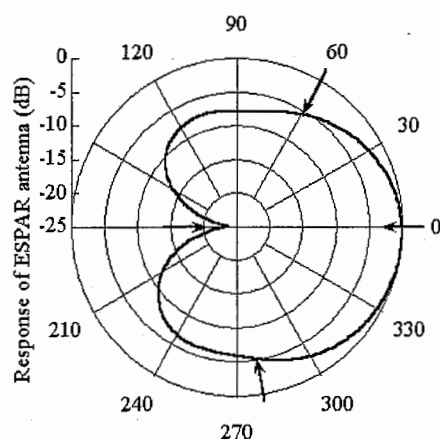


Figure 3.6: ESPAR antenna pattern

Figure 3.7 shows a QPSK constellation of the received signals after beamforming and OFDM demodulation and shows that the data can be perfectly recovered by the proposed receiver. This simulation shows that the ESPAR antenna can work in an OFDM scheme to suppress the delayed paths out of the GI.

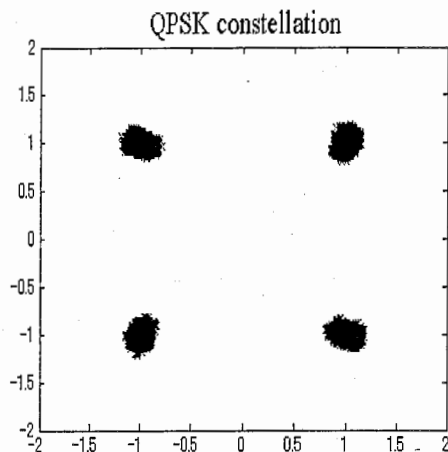


Figure 3.7: Depressed signals before decision

For the third simulation we now assume that three of the five paths are out of the GI (see Table 3.3) and obviously cannot be canceled only by the OFDM demodulator with an omni-directional antenna. Figure 3.8 shows that these 3 path signals are efficiently canceled by using the ESPAR antenna in an OFDM scheme (see Figure 3.3).

Table 3.3: Third simulation assumptions

	Impinging signals				
	1	2	3	4	5
DoA	0°	150°	60°	260°	300°
Delay (samples)	0	5	17	21	25
Power (normalized)	1	0.5	0.5	0.5	0.5

When creating the receiver, we can make the ESPAR antenna cancel only the delayed paths that cannot be suppressed by the OFDM demodulator. The QPSK constellation of the received signals is shown in Figure 3.9 and illustrates that in this case too the data can be recovered. So the proposed receiver is sufficiently efficient to cancel several long-time delayed paths out of the GI.

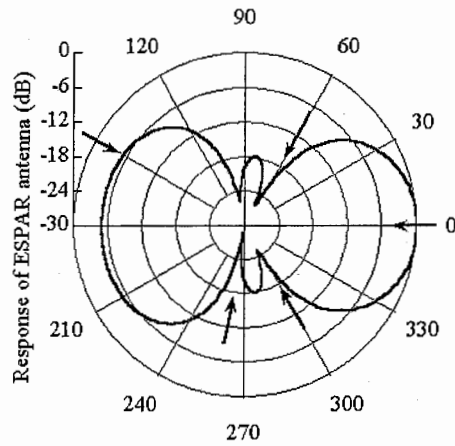


Figure 3.8: ESPAR antenna pattern

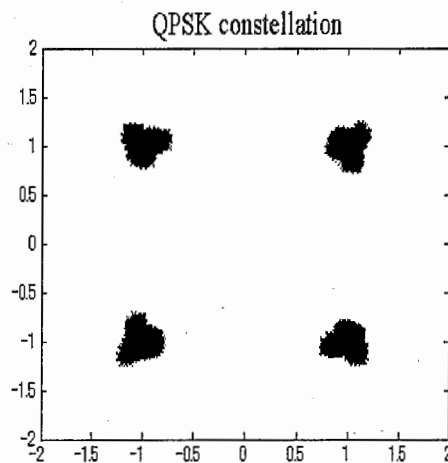


Figure 3.9: Depressed signals before decision

### 3.4.3 Performances in term of BER

In the fourth simulation, Figure 3.10 shows the BER according to the number of delayed copies of the main path that are out of the GI. The solid curve is the BER for 3 delayed paths with 2 out of the GI. The curve with black dots is the BER for 4 delayed paths with 3 out of the GI. We use as a reference the dashed curve, which is the theoretical BER curve for QPSK assuming there are no delayed copies of the main path, no interference and only noise with an omni-directional antenna, with the data in QPSK.

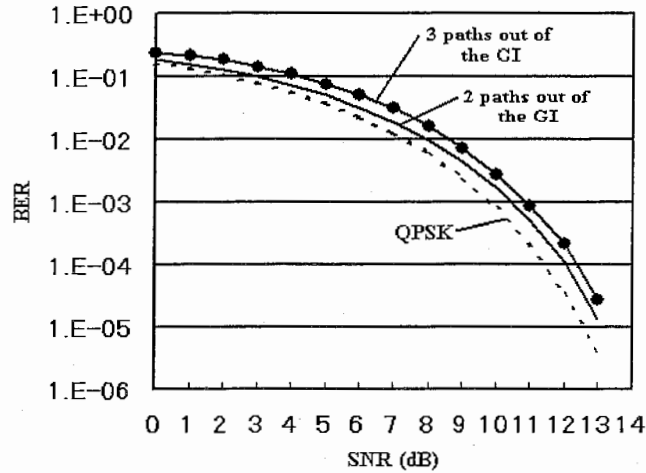


Figure 3.10: BER versus the number of delayed paths

There is a difference less than one dB between the theoretical curve and the solid curve and less than two dB between the dashed curve and the curve with black dots. So even with 2 or 3 delayed paths out of the GI, we significantly prevent the decrease in performance thanks to the ESPAR antenna used in an OFDM scheme.

### 3.5 Conclusion

The proposed combination of the ESPAR antenna and OFDM requires only one FFT and has low complexity thanks to the simple structure of the ESPAR antenna. Whereas the performance degrades drastically when delayed paths are out of the GI, using the ESPAR antenna greatly improves the BER performance by effectively suppressing the delayed signals out of the GI even with long-time delays. Simulations showed that for 4 impinging delayed paths in which 3 paths are out of the GI, we can reach a BER of less than  $10^{-4}$  for a SNR of 13 dB, which is close to the theoretical case with only the main path impinging. With the good performance shown by the simulations and the low complexity of the proposed system, the ESPAR antenna used in an OFDM scheme is considered to be a suitable candidate for user terminals.

## Chapter 4

# Reactance-domain MUSIC algorithm for ESPAR antenna

### 4.1 Introduction

The determination of the parameters of multiple wavefronts impinging on an array antenna is a significant issue in digital array processing. The Multiple Signal Classification (MUSIC) algorithm [10] is currently used in conventional array antenna processing to provide asymptotically unbiased estimates of the characteristics of incident signals, like the number of signals, the directions-of-arrivals and the strengths and cross correlations among the directional waveforms.

Mobile user terminals have stringent limitations on hardware and algorithm complexity as well as power consumption. It is well known that nearly all-existing antenna array techniques also require one receiver chain per branch of antenna. As the ESPAR antenna [3] has a single-port output, it requires little hardware and so the ESPAR antenna features low consumption and low cost. Thus the ESPAR antenna is well suited for application to wireless communications systems, and especially to mobile user terminals.

Because of the configuration [3] of the ESPAR antenna, we may face some difficulties in applying conventional algorithms to this antenna. The antenna pattern is formed according to the values of the loaded reactances on the parasitic radiators. So the signals on the parasitic elements, which surround the active element, cannot be observed. Only the single-port output can be observed and processed. The RF currents on the elements are not independent but mutually coupled with each other, and depend on the values of the reactances. Plus the single-port output is a highly nonlinear function of these reactances. All of these characteristics prevent conventional algorithms for being applicable to the ESPAR antenna.

The previous DoA finder [11] used with the ESPAR antenna is based on a switched beam method, in which a beam is switched to find the direction

where the received power is maximum. But this method has only a resolution of 30 degrees and a precision of 15 degrees. In addition, we can estimate the DoA of only one impinging signal. So the purpose of this chapter is to develop a DoA finder that can estimate the DoAs of several impinging signals with better resolution and precision.

To realize this aim, we propose a reactance-domain MUSIC algorithm based on the conventional MUSIC algorithm. The conventional MUSIC algorithm needs to compute the correlation matrix of the output vector of the antenna elements. But since we can observe only one element on the ESPAR antenna, we have to propose a method to create a correlation matrix for a single-port output antenna. The proposed solution is based on the reactance diversity of the ESPAR antenna. The goal is to recreate the spatial diversity of a conventional array based on the variable reactances connected to the parasitic elements.

Since a correlation matrix is available, we can apply a modified version of the conventional MUSIC algorithm to this correlation matrix. Then we propose a reactance-domain MUSIC algorithm to estimate the DoAs and the parameters of several incident signals.

## 4.2 Correlation matrix of a single-port antenna

For the ESPAR antenna, the signals on the surrounding passive elements (#1 – # $M$ ) cannot be observed. Only the single-port output can be observed and processed. The RF currents on the elements are not independent but mutually coupled with each other, and depend on the values of the reactances. The single-port output is a highly nonlinear function of these variable reactances. In conventional adaptive arrays, the correlation matrix is created by measuring the signals on each element of the antenna, so the problem is the same for all kinds of single-port output antennas: how to create a correlation matrix with only one output port ?

The goal is to recreate with a single-port output antenna the spatial diversity of a conventional array antenna.

First we consider that we can monitor the  $M$  elements of a conventional array antenna in the same time and compute the correlation matrix. This is the normal case (see Figure 4.1(a)).

Now we consider only a single element antenna. If we want to recreate the spatial diversity available in a conventional array antenna with  $M$  elements, we must change the position of our single element antenna and measure the output  $M$  times by sending the same signals each time (see Figure 4.1(b)). This way we can obtain the  $M$  output values as if we were using an  $M$ -element antenna. In fact, we use a decomposition in time to



recreate the spatial diversity of the  $M$ -element antenna. So we now have all of the output needed to compute the correlation matrix. We can conduct the same reasoning with the single-port output ESPAR antenna but we will now use the reactance-domain to recreate the spatial diversity.

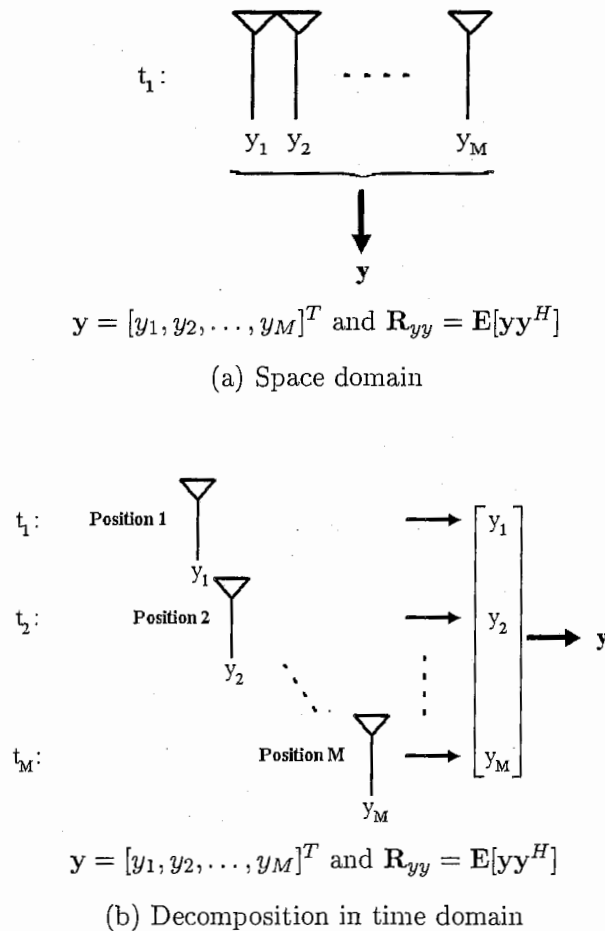


Figure 4.1: Correlation matrix for conventional array antenna

The ESPAR antenna possesses  $M$  variable reactances. So first we set a reactance vector and measure the output. However we can always measure only the same element, so to recreate the spatial diversity we change the set of reactances and we measure the output again by sending the same signals, and by repeating this process  $M$  times we are able to get the  $M$  outputs needed to create a correlation matrix (see Figure 4.2).

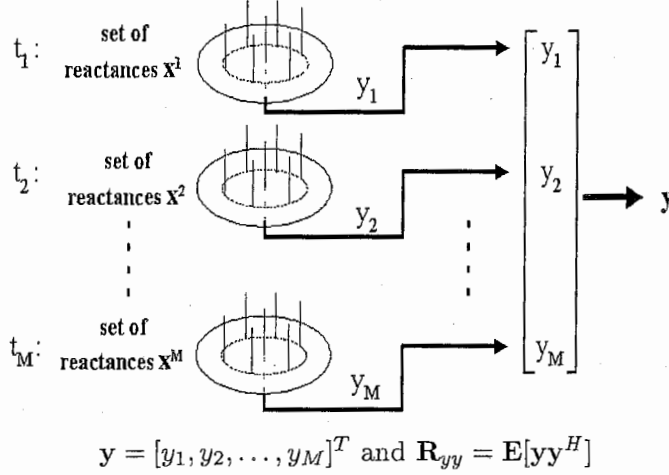


Figure 4.2: Correlation matrix in reactance domain

### 4.3 Reactance-domain MUSIC algorithm

The waveforms received by the ESPAR antenna elements are combinations of the incident wavefronts and noise. For the ESPAR [3] antenna we are in the reactance domain. So to compute the output  $y_m$  (see Figure 4.2), we first use the set of reactances  $\mathbf{x}^m$  to calculate the equivalent current vector  $\mathbf{i}_m$  and then we can get the output  $y_m$ . By changing the values of the reactances, and so the pattern of the antenna and the current vector  $M$  times, we are able to recreate the spatial diversity available in the conventional case and finally we can know the  $\mathbf{y}$  output vector used to compute the correlation matrix. The  $M$  measured output can be expressed as

$$\begin{aligned}
 y_1(t_1) &= \sum_{d=1}^D \mathbf{i}_1^T \mathbf{a}(\theta_d) u_d(t_1) + n(t_1) \\
 y_2(t_2) &= \sum_{d=1}^D \mathbf{i}_2^T \mathbf{a}(\theta_d) u_d(t_2) + n(t_2) \\
 &\vdots \\
 y_M(t_M) &= \sum_{d=1}^D \mathbf{i}_M^T \mathbf{a}(\theta_d) u_d(t_M) + n(t_M)
 \end{aligned} \tag{4.1}$$

under the assumption that  $u_d(t_1) = u_d(t_2) = \dots = u_d(t_M)$  for  $d = 1, \dots, D$ , which is equivalent to saying that the same signals are sent repetitively  $M$  times. This leads to the  $\mathbf{y}$  output vector, which can be expressed as follows, where  $t$  is omitted for the sake of simplicity,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \mathbf{i}_1^T \\ \mathbf{i}_2^T \\ \vdots \\ \mathbf{i}_M^T \end{bmatrix} \begin{bmatrix} \mathbf{a}(\theta_1) & \dots & \mathbf{a}(\theta_D) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_D \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_M \end{bmatrix}$$

or

$$\mathbf{y} = \mathbf{I}^T \mathbf{A} \mathbf{u} + \mathbf{n} \quad (4.2)$$

The incident signals are represented in amplitude and phase by the complex vector  $\mathbf{u} = [u_1, u_2, \dots, u_D]^T$  and the spatial noise appears as the complex vector  $\mathbf{n} = [n_1, n_2, \dots, n_M]^T$ . The elements of  $\mathbf{y}$ ,  $\mathbf{I}^T$  and  $\mathbf{A}$  are also complex in general. The  $\mathbf{A}$  matrix is composed by  $D$  columns, each column is the steering vectors  $\mathbf{a}(\theta_j)$  corresponding to the response to the direction of arrival  $\theta_j$  of the  $j$ th signal. These steering vectors are known functions depending on the structure of the antenna. The  $\mathbf{I}^T$  matrix is composed by  $D$  lines, each line is a complex current vector compute based on the knowledge of the antenna and which depends on the reactance vector.

Lets denote

$$\mathbf{a}_{mod}(\theta) = \mathbf{I}^T \mathbf{a}(\theta) \quad (4.3)$$

as the modified steering vector for the reactance-domain MUSIC algorithm.

We can now rewrite Eq. (4.2) as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{mod}(\theta_1) & \dots & \mathbf{a}_{mod}(\theta_D) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_D \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_M \end{bmatrix}$$

and under this shape, this equation is similar to the one used in the conventional MUSIC algorithm [10].

Now, we can give some explanations on the reactance-domain MUSIC algorithm (see Figure 4.3).

The estimation of the  $M$  by  $M$  correlation matrix of the  $\mathbf{y}$  vector is given by

$$\mathbf{R}_{yy} = \mathbf{E}[\mathbf{y}\mathbf{y}^H], \quad (4.4)$$

where the superscript  $H$  is the transpose and conjugate operation.

If the number of incident wavefronts is less than the number of parasitic elements  $M$  and under the assumption that the incident signals and the noise are uncorrelated, the matrix  $\mathbf{F} = \mathbf{E}[\mathbf{u}\mathbf{u}^H]$  is diagonal or may contain

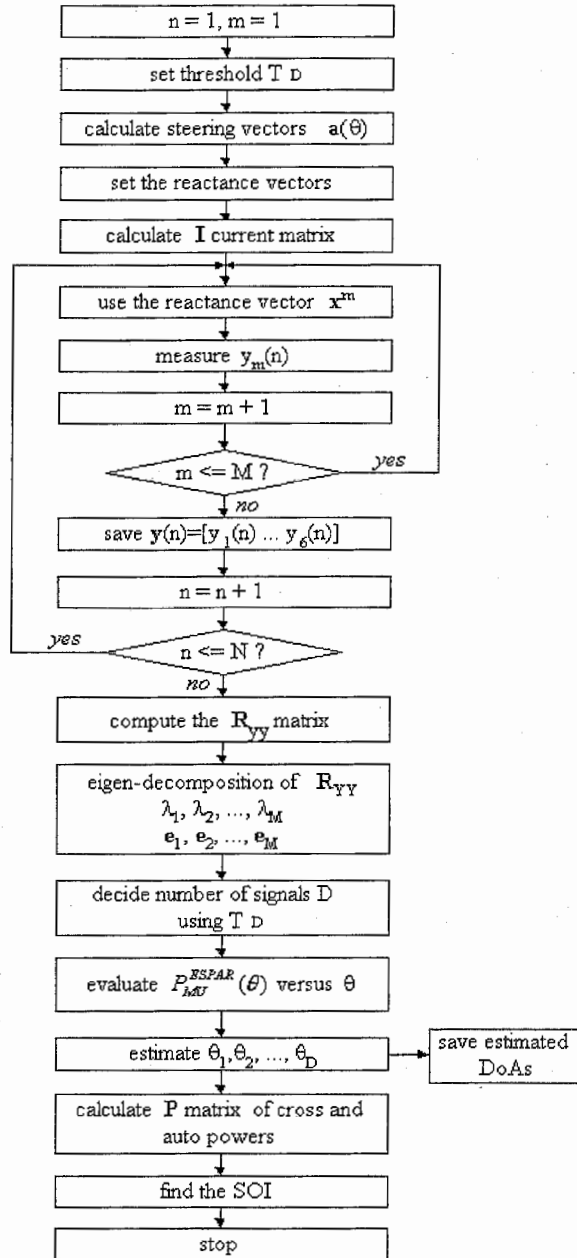


Figure 4.3: Flowchart of the reactance-domain MUSIC algorithm

completely correlated pairs ( $\mathbf{F}$  is singular). In general,  $\mathbf{F}$  will be “merely” positive definite and if the noise vector  $\mathbf{n}$  is mean zero, variance  $\sigma^2$ , the Eq. (4.4) can be rewritten as

$$\mathbf{R}_{yy} = \mathbf{I}^T \mathbf{A} \mathbf{F} \mathbf{A}^H \mathbf{I}^* + \lambda_{min} \mathbf{I}_M. \quad (4.5)$$

Now we can make the eigen decomposition of the  $\mathbf{R}_{yy}$  matrix and get the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_M$  ( $\lambda_{min}$  is the smallest eigenvalue) and the associated eigenvectors  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_M$ . The rank of  $\mathbf{I}^T \mathbf{A} \mathbf{F} \mathbf{A}^H \mathbf{I}^*$  is  $D$ , the number of incident signals. So we can estimate the number of incident signals by choosing from the eigenvalues ( $\lambda_i$  for  $i = D + 1, \dots, M$ ) the highest values determined by a threshold  $T_D$ . The number of eigenvalues which are chosen determine the estimated number of impinging signals  $D$ . In the complete set of eigenvalues of  $\mathbf{R}_{yy}$ ,  $\lambda_{min}$  will not always be simple, in fact it occurs repeated  $N = M - D$  times. Therefore, the number of incident signals estimator is  $D = M - N$  where  $N$  is the multiplicity of  $\lambda_{min}$ .

The  $D$  eigenvectors associated to the  $D$  greater eigenvalues of  $\mathbf{R}_{yy}$  generate the signal subspace  $\mathbf{E}_S$  and the  $N = M - D$  eigenvectors generate the noise subspace  $\mathbf{E}_N$ .  $\mathbf{R}_{yy}$  is hermitian, thus  $\mathbf{E}_S$  and  $\mathbf{E}_N$  are orthogonal. The modified steering vectors of the incident signals generate the signal subspace  $\mathbf{E}_S$  too. Finally the goal is to find the DoAs which minimize the modified steering vectors projections on the noise subspace  $\mathbf{E}_N$ .

If  $\mathbf{E}_N$  is defined to be the  $M$  by  $N$  matrix whose columns are the  $N$  noise eigenvectors, and the ordinary Euclidian distance from a vector  $\mathbf{a}_{mod}(\theta)$  belonging to the modified steering vector (cf. Eq. (4.3)) continuum is  $d^2 = \mathbf{a}_{mod}^H(\theta) \mathbf{E}_N \mathbf{E}_N^H \mathbf{a}_{mod}(\theta)$ . So we can plot  $\frac{1}{d^2}$  for points along the  $\mathbf{a}_{mod}(\theta)$  continuum as a function of  $\theta$ . that is

$$P_{MU}^{ESPAR}(\theta) = \frac{1}{\mathbf{a}(\theta)^H \mathbf{I}^* \mathbf{E}_N \mathbf{E}_N^H \mathbf{I}^T \mathbf{a}(\theta)} = \frac{1}{\mathbf{a}_{mod}(\theta)^H \mathbf{E}_N \mathbf{E}_N^H \mathbf{a}_{mod}(\theta)} \quad (4.6)$$

for  $0^\circ \leq \theta < 360^\circ$  and where  $\mathbf{E}_N = [\mathbf{e}_{D+1}, \mathbf{e}_{D+2}, \dots, \mathbf{e}_M]$ .

It is easy now to estimate the DoAs of the impinging signals. We must only find the values on the x-axis of the peaks on the MUSIC spectrum to find the estimation of the incident signals. Now we can know the number of impinging signals and their estimated DoAs and we can use this information to learn the power of the incident signal. It is very interesting to determine which signal is the signal of interest (SOI) under the assumption that it is the more powerful signal. Since we know the estimated DoAs, the  $\mathbf{A}$  matrix becomes available and may be used to compute the parameters of the incident signals contained in the  $D$  by  $D$   $\mathbf{P}$  matrix of cross and auto powers of the incident signals. Then the  $\mathbf{P}$  matrix can be computed as

$$\mathbf{P} = (\mathbf{A}^H \mathbf{I}^* \mathbf{I}^T \mathbf{A})^{-1} \mathbf{A}^H \mathbf{I}^* (\mathbf{R}_{yy} - \lambda_{min} \mathbf{I}_M) \mathbf{I}^T \mathbf{A} (\mathbf{A}^H \mathbf{I}^* \mathbf{I}^T \mathbf{A})^{-1}, \quad (4.7)$$

where  $\mathbf{I}_M$  is the  $M$  by  $M$  unit matrix and  $\lambda_{min}$  is the smallest eigenvalue of the correlation matrix. The superscript  $*$  is the conjugate operation. This

is almost the same expression as in the conventional MUSIC algorithm [10], the only modification being that we replace the  $\mathbf{A}$  matrix by the  $\mathbf{I}^T \mathbf{A}$  matrix. The diagonal of this  $\mathbf{P}$  matrix contains the estimated power of each incident signal, so it is easy to find the signal of interest under the assumption that it is the more powerful.

#### 4.4 Computer simulations with MATLAB

In all of the following simulations, we assume that all of the impinging signals have the same carrier frequency, and the data clock synchronisation is complete. We also neglect the Doppler effect.

The first simulation is an example showing concretely how to get the estimated DoAs from the MUSIC spectrum. For this example we use six different reactance vectors randomly chosen. Figure 4.4 shows the MUSIC spectrum and the values of the estimated DoAs for four incident signals, each one with a SNR of 5 dB. In fact the x-axis values of the four maximum peaks give the estimation of the DoAs. The peaks are sharp and thus the precision of the reactance-domain MUSIC algorithm is about 0.5 degree. So this simulation shows that the proposed reactance-domain MUSIC algorithm can work to estimate the DoAs of impinging signals on the ESPAR antenna.

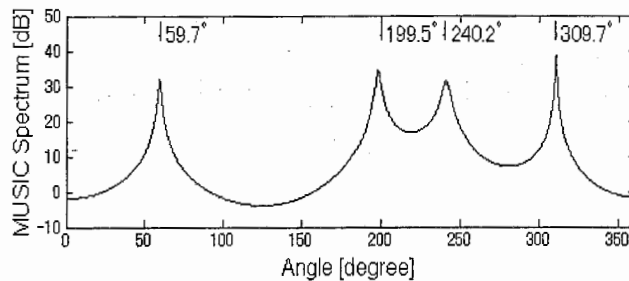


Figure 4.4: MUSIC spectrum for random set of reactances, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal

In the second simulation, we want to investigate the resolution of the reactance-domain MUSIC algorithm. We will consider only two incident signals but with close DoAs to estimate the maximum resolution that can be reached by our proposed algorithm. The SNR of the two signals is 10 dB. Figure 4.5 shows the MUSIC spectrum where the two signals are separated by only 10 degrees. This is the limit case where we can still estimate the two DoAs with good precision. Finally, the resolution of the proposed reactance-domain MUSIC algorithm is about 10 degrees.

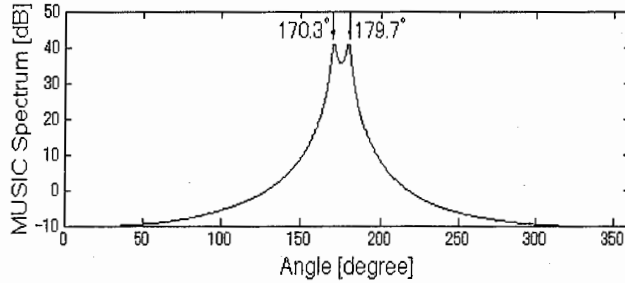


Figure 4.5: MUSIC spectrum for random set of reactances, DoAs=170°, 180°, SNR=5 dB for each signal

The third simulation is conducted in order to determine the performance of the ESPAR antenna using the reactance-domain MUSIC algorithm. For this reason, we make a comparison with a conventional antenna having the same structure as the ESPAR antenna [12, 13] (see Figure 4.6).

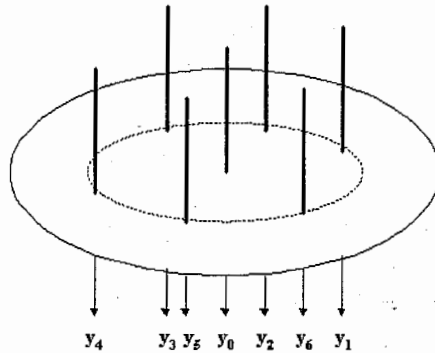


Figure 4.6: Circular antenna with mutual coupling

The radius is the same as that for the ESPAR antenna and is equal to  $\frac{\lambda}{4}$ . This radius is smaller than  $\frac{\lambda}{2}$  so some mutual coupling effects are introduced as with the ESPAR antenna. To deal with this mutual coupling, we must introduce the  $\mathbf{C}$  matrix, which plays the same role as the  $\mathbf{I}$  matrix for the ESPAR antenna. The  $\mathbf{C}$  matrix can be expressed as  $\mathbf{C} = \mathbf{Z}_S(\mathbf{Z}_S + \mathbf{Z})^{-1}$ , where  $\mathbf{Z}_S$  is a diagonal matrix filled with the impedance value of 50  $\Omega$ , and  $\mathbf{Z}$  is the impedance matrix used for the ESPAR antenna (*cf.* section 2.2.2). Figure 4.7 shows the MUSIC spectrum for the conventional array described above and the curve obtained for the reactance-domain MUSIC algorithm (*cf.* first simulation). The two curves are very close. Furthermore, the estimated DoAs are in the same range for the two antennas. So with the reactance-domain MUSIC algorithm we can reach the same performance as in the conventional case in resolution and precision.

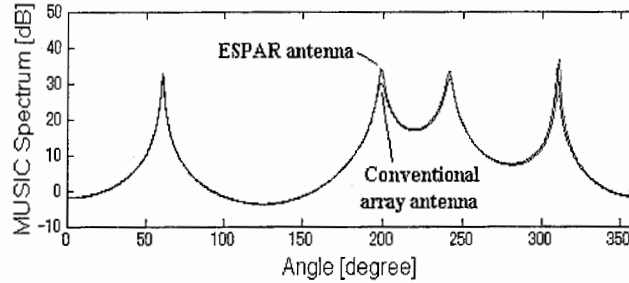


Figure 4.7: MUSIC spectrum comparison between conventional array antenna and ESPAR antenna, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal

The fourth simulation will deal with further practical application of the reactance-domain MUSIC algorithm. For each set of reactances we can use random values, but the structure of the ESPAR antenna is symmetric and because of the calibration of this antenna, it is easier in practice to use one set of reactances and to shift the values of the reactances each time we want to measure an output.

Moreover, with this shifted method the needed hardware will be easier to realize. So we will investigate this method to verify whether we can get the same performance as in the random case. Figure 4.8 shows the MUSIC spectrum obtained with the shifted reactance method and the curve obtained with the random method (*cf.* first simulation). The curves are very close, and so the resolution and the precision for the estimated values of DoAs are the same for the two methods. The resolution is about 10 degrees and the precision about 0.5 degree. Thus, we will use this shifted reactance method in the following simulations to fit with practical cases.

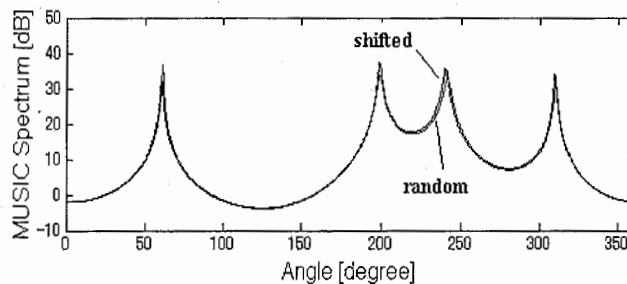


Figure 4.8: MUSIC spectrum comparison between shifted method and random method for the set of reactance, DoAs=60°, 200°, 240°, 310°, SNR=5 dB for each signal

In the fifth simulation, we want to show the influence of the incident



signal SNR on the MUSIC spectrum. So we will show and discuss the results for a SNR of 20 dB, 5 dB and 0 dB. The case with 5 dB was already done in the first simulation. Figure 4.9 shows the MUSIC spectrum for these respective SNR. It is clear by looking at the curves that when the SNR increases, the peaks become sharper and higher, thus the resolution becomes better. As we can see on the estimated DoAs (see Table 4.1), the precision also improves when the SNR increases. But it is interesting to note that even with a SNR of 0 dB we obtain good results in resolution and especially in precision. The precision is about 0.5 degree and is the same as when the SNR is 5 dB.

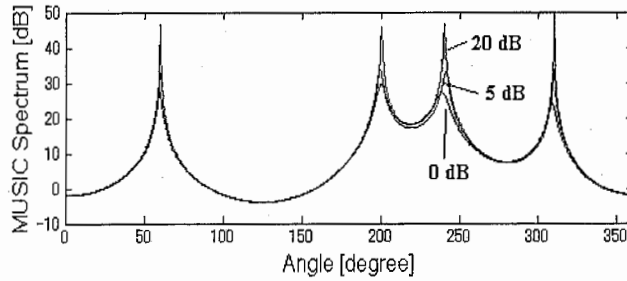


Figure 4.9: MUSIC spectrum for SNR=20, 5 and 0 dB, DoAs=60°, 200°, 240°, 310°

Table 4.1: DoA estimation for different SNR

	Impinging signals			
	1	2	3	4
DoA	60°	200°	240°	310°
Estimated DoAs (SNR=20 dB)	60.0°	200.1°	240.0°	299.9°
Estimated DoAs (SNR=5 dB)	59.7°	199.5°	240.3°	309.7°
Estimated DoAs (SNR=0 dB)	59.6°	199.5°	239.8°	309.5°

In the sixth simulation, the goal is to determine which signal is the SOI. This SOI is the signal that we want to find among all the other impinging signals. The SOI is assumed to be more powerful than the other incident signals (see Table 4.2). This assumption is frequently made in practice. Since the MUSIC spectrum does not give enough information on the power of the incident signals, we can estimate the respective power of each signal by computing the  $\mathbf{P}$  matrix of the cross and auto power (*cf.* Eq. (4.7)). The

diagonal of the  $\mathbf{P}$  matrix gives the estimated power of each impinging signal.

$$\mathbf{P} = \begin{bmatrix} 4.0235 & 0.0175 & 0.0202 & 0.0106 \\ 0.0175 & 3.1483 & 0.0159 & 0.0200 \\ 0.0202 & 0.0159 & 8.0149 & 0.0109 \\ 0.0106 & 0.0200 & 0.0109 & 4.9976 \end{bmatrix}$$

So the position of the highest value in the  $\mathbf{P}$  diagonal corresponds to the number of the SOI in Table 4.2. The computed  $\mathbf{P}$  matrix shows that in this simulation the incident signals were almost uncorrelated because the values outside the diagonal are very small. The third diagonal value 8.0149 is the highest, so we find that the SOI is the third incident signal (see Table 4.2) with an estimated DoA of 180.3 degrees. So by computing the  $\mathbf{P}$  matrix, thanks to the estimated DoAs obtained by the MUSIC spectrum, we are able to determine the SOI among all of the incident signals under the assumption that it is the more powerful signal.

Table 4.2: DoA estimation for the signal of interest

	Impinging signals			
	1	2	3 (SOI)	4
SNR (dB)	3	2	6	4
DoA	10°	110°	180°	300°
Estimated DoAs	10.2°	109.6°	180.3°	300.1°

In the above simulations, according to the reactance-domain MUSIC algorithm, all of the incident signals are sent periodically in order to compute the correlation matrix  $\mathbf{R}_{yy}$  (*cf.* Eq. (4.4)).

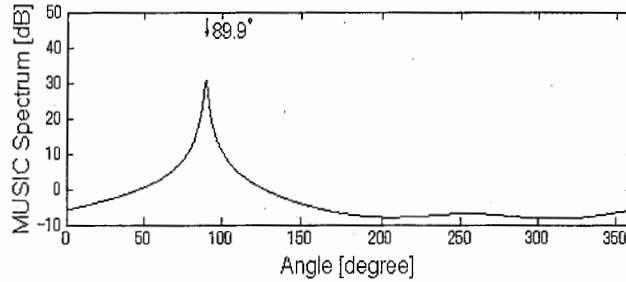


Figure 4.10: MUSIC spectrum for periodic SOI (DoA=90°) and 3 no periodic interferences (DoAs=150°, 210° and 300°)

In the seventh simulation, we will consider the case where only the SOI (*cf.* sixth simulation) is sent periodically to the ESPAR antenna. The other

signals are interferences supposed to come from unknown sources and so are not sent periodically. The MUSIC spectrum (see Figure 4.10) shows that even if we cannot estimate the DoAs of the interferences, we can estimate the DoA of the SOI with the same precision as in the previous simulations.

The eighth simulation shows the behavior of the reactance-domain MUSIC algorithm when the impinging signals are composed of a main path and several delayed version of this path (see Table 4.3), when the delayed path are supposed to be incoherent, *i.e.*, the delays are greater than one sample duration. The main path is assumed to be more powerful than the reflected and delayed paths. This assumption is made because we want to know which of the incident paths is the main path. Figure 4.11 shows the MUSIC spectrum for our delayed paths.

$$\mathbf{P} = \begin{bmatrix} 7.8999 & 2.5706 & 0.5552 & 0.1312 \\ 2.5706 & 5.0187 & 0.7555 & 0.2915 \\ 0.5552 & 0.7555 & 4.9942 & 0.7803 \\ 0.1312 & 0.2915 & 0.7803 & 5.0575 \end{bmatrix}$$

By searching for the highest value in the diagonal of the  $\mathbf{P}$  matrix we can determine which impinging path is the main path that we want to recover. In this simulation, the highest value is the first value of the diagonal and is equal to 7.8999, thus the main path is the number 1 path in Table 4.3 and its estimated DoA is 49.9 degrees. We can confirm that the reactance-domain MUSIC algorithm behavior in a multipath environment is good because the peaks are sharp and high and because we reach a good precision of 0.5 degree for the estimated DoAs.

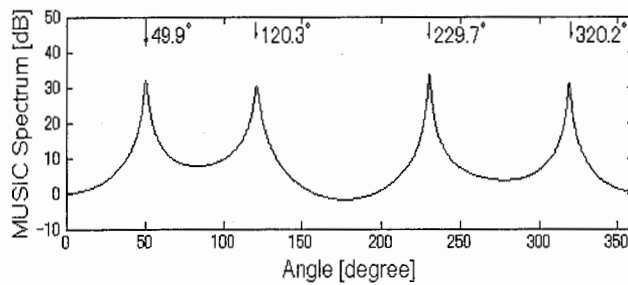


Figure 4.11: MUSIC spectrum in multipath environment, DoAs=50°, 120°, 230°, 320°

Remark: if the delays are smaller than one sample duration, the impinging paths are correlated and thus the reactance-domain MUSIC algorithm fails as in the conventional case.

Table 4.3: DoA estimation in multipath environment

	Impinging signals			
	1	2	3	4
SNR (dB)	6	4	4	4
Delays (samples)	0	1	4	7
DoA	50°	120°	230°	320°
Estimated DoAs	49.9°	120.3°	229.7°	320.2°

In the ninth simulation, we will investigate the influence of the range of reactances (*cf.* section 2.2.2). For the above simulations, the range of reactances was set to  $[-300 \Omega$  to  $300 \Omega]$ , but in practical cases the available range of reactances is  $[-93.6 \Omega$  to  $-4.8 \Omega]$ , so we want to check whether if the reactance-domain MUSIC algorithm can maintain the same performance even with a narrower range of reactances.

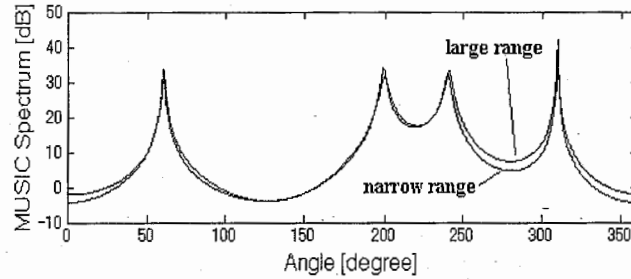


Figure 4.12: MUSIC spectrum for narrow range of reactances, DoAs=60°, 200°, 240°, 310°

Table 4.4: DoA estimation for large and narrow range of reactance

	Impinging signals			
	1	2	3	4
SNR (dB)	5	5	5	5
DoA	60°	200°	240°	310°
Estimated DoAs large	59.7°	199.5°	240.2°	309.7°
Estimated DoAs narrow	59.8°	199.5°	240.3°	309.8°

Figure 4.12 shows the MUSIC spectrum obtained for the narrow range of reactances and for the previous range of reactances. Thus we can verify that the MUSIC spectrum obtained for the two ranges of reactances are

almost the same. Plus the DoAs are estimated with the same precision in these two cases (see Table 4.4). Thus we can say that even with the range of reactances used in practice we can get the same performance in resolution and precision than if we were using a larger set of reactances.

In the above simulations, we successfully computed the correlation matrix  $\mathbf{R}_{yy}$  (cf. Eq. (4.4)) for  $N=1000$  (see Figure 4.3), so the total number of samples used in the reactance-domain algorithm was 6000. The aim of this last simulation is to show the influence of the number of samples used to estimate the correlation matrix. Figure 4.13 shows the results for  $N = 1000$  and  $N = 50$ , that is to say that we use only 300 samples to estimate the correlation matrix. This issue is very important because in practical cases we want to decrease the calculation cost as much as possible by using the minimum amount of samples to get good results. Figure 4.13 shows clearly that when the number of samples decreases, the peaks become smaller and less sharp, so the resolution decreases, but if we check the estimated DoAs (see Table 4.5), we can notice that the precision of the estimated DoAs remains good, that is to say, about 0.5 degree. So the reactance-domain MUSIC algorithm can work and provide a valuable estimation of DoAs even with a few samples.

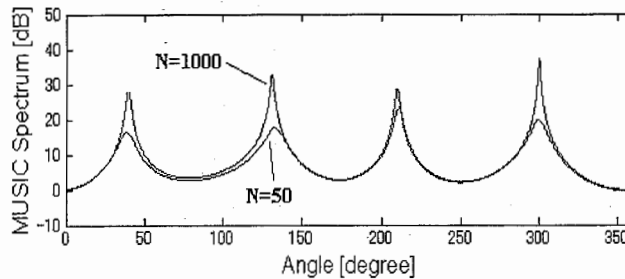


Figure 4.13: MUSIC spectrum for  $N=1000$  and  $N=50$ , DoAs= $40^\circ$ ,  $130^\circ$ ,  $210^\circ$ ,  $300^\circ$

Table 4.5: Influence of the number of samples on the DoA estimation

	Impinging signals			
	1	2	3	4
SNR (dB)	5	5	5	5
DoA	$40^\circ$	$130^\circ$	$210^\circ$	$300^\circ$
Estimated DoAs 6X1000	$40.1^\circ$	$129.6^\circ$	$209.6^\circ$	$300.3^\circ$
Estimated DoAs 6X50	$39.7^\circ$	$129.5^\circ$	$209.8^\circ$	$299.6^\circ$

## 4.5 Discussions for practical cases

It is important to notice that the antenna pattern is modified when we change the reactances values. For each set of reactances we can use random values but the efficiency of the reactance-domain MUSIC algorithm depends on the chosen reactances. Some set of reactances can lead to lower performances than the ones described in the simulations. Thus it is very important to choose carefully the set of reactances used in our proposed algorithm.

The current matrix  $\mathbf{I}^T$  is introduced to take in account the beamforming of the ESPAR antenna created by the reactance vector. Notice that in the conventional case we don't make the antenna form beams and as a consequence this matrix is not present in the conventional MUSIC algorithm. This matrix is computed based on the knowledge of the impedance  $\mathbf{Z}$  matrix (*cf.* section 2.2.2). In practical cases, the  $\mathbf{Z}$  matrix which characterize the mutual coupling for the ESPAR antenna is not perfectly known, this is a problem of calibration of the antenna. So, if this matrix does not fit perfectly with the ESPAR antenna, the efficiency of the proposed algorithm can be decreased. But in practice we can face this problem when we want to estimate the DoA of one incident signal. With only one impinging signal, measured in anechoic chamber for example, and by neglecting the noise, the output of the ESPAR antenna (Eq. (2.15)) can be simply rewritten as

$$\mathbf{y}(\theta) = \mathbf{I}^T \mathbf{a}(\theta) u(t) \quad (4.8)$$

Now if we set the signal  $u(t)$  to the constant 1, the output expression of the ESPAR antenna becomes

$$\mathbf{y}(\theta) = \mathbf{I}^T \mathbf{a}(\theta) = \mathbf{a}_{mod}(\theta). \quad (4.9)$$

Notice that by plotting the squared module of  $y$  for all the values of  $\theta$  we obtain the pattern of the ESPAR antenna. This way there is no need to know the  $\mathbf{Z}$  matrix because we can use the measured values of the pattern directly in the expression of the MUSIC spectrum as following

$$P_{MU}^{ESPAR}(\theta) = \frac{1}{\mathbf{a}_{mod}(\theta)^H \mathbf{E}_N \mathbf{E}_N^H \mathbf{a}_{mod}(\theta)} \quad (4.10)$$

So in this case the MUSIC spectrum can be known without the errors introduced by the not perfectly known  $\mathbf{Z}$  matrix.

## 4.6 Conclusion

We have proposed in this chapter a reactance-domain MUSIC algorithm for the ESPAR antenna. The simulations show that our proposed algorithm

can be used with an ESPAR antenna to estimate the DoAs of impinging signals. The comparison of this algorithm and the conventional MUSIC algorithm shows that even if the ESPAR antenna is a single-port output we can reach the same performance than in conventional cases. The precision of the reactance-domain MUSIC algorithm is less than 0.5 degree and the resolution is less than 10 degrees.

The resolution and the precision of our algorithm become better as the SNR increases but it is interesting to note that this algorithm exhibits a good behavior even for very small values of SNR, like 0 dB, so that we can keep the same precision, 0.5 degree, as in the case where the SNR is 5 dB.

The reactance-domain MUSIC algorithm provides a valuable estimation of the impinging signal DoAs and by computing the  $\mathbf{P}$  matrix of the cross and auto power, we are able to determine the signal of interest among all of the incident signals under the assumption that it is the more powerful signal. In fact, the diagonal of the  $\mathbf{P}$  matrix gives the estimated power of each incident signal.

The case where only the SOI is sent periodically to the ESPAR antenna and where the other signals are interferences supposed to come from unknown sources and so are not sent periodically can occur. In this case, even if we cannot estimate the DoAs of the interferences, we can still estimate the DoA of the SOI with the same precision as described above.

The reactance-domain MUSIC algorithm can be used in a multipath environment under the assumption that the delays are greater than a sample duration. Its behavior is good and we can reach a good precision of 0.5 degree for estimated DoAs with a SNR of about 5 dB for the incident signals.

For each set of reactances we can use random values, but because of the structure of the ESPAR antenna and because of the hardware to implement our algorithm, it is better to use one set of reactances and to shift the values of the reactances each time we want to measure an output. The simulations have confirmed that the obtained results are as good with the shifted method as with the random method.

Now we can investigate future applications of the reactance-domain MUSIC algorithm in experiments. In practice, the range of reactances that can be used for the ESPAR antenna is smaller than that used in simulation, but even with the narrow range used in practice we can get the same performance in resolution and precision than if we were using a larger set of reactances.

An important issue is the number of samples necessary to estimate the correlation matrix. When the number of samples decreases, the resolution decreases, but the precision of the estimation remains good, that is to say, about 0.5 degree. So the reactance-domain MUSIC algorithm can work and provide a valuable estimation of DoAs even with a few samples, like 300, as displayed in our last simulation. Finally, the reactance-domain MUSIC algorithm, is truly a good candidate for estimating the signal DoAs impinging

on the ESPAR antenna.

Now that we have found a way to compute a correlation matrix in the reactance domain for single-port antennas, many investigations can be done to adapt the conventional methods using a cross correlation matrix to the family of single-port output antennas in the same way as the reactance-domain MUSIC algorithm for the ESPAR antenna.



## Conclusion

The WACNet is considered as a means of communications among portable user terminals that temporarily meet, where distance and time come close yet easy connection to a network infrastructure is not possible. One of the key technologies developed to achieve the WACNet is the ESPAR antenna embedded in user terminals. The ESPAR antenna has been proposed as a practical hardware realization of the ABF concept. The main issue is that ESPAR antenna is not a conventional array antenna. Then, theories about conventional array antennas cannot be directly applied. So to improve the performance of the WACNet we proposed two new ways to take advantages from the ESPAR antenna.

On the one hand, we proposed a combination of the ESPAR antenna and OFDM modulation which requires only one FFT device. The good performance shown by the simulations and the low complexity of the proposed system make the ESPAR antenna used in an OFDM scheme to be a suitable candidate for user terminals to fit to the WACNet exigencies.

On the other hand we proposed a reactance-domain MUSIC algorithm for the ESPAR antenna. Our proposed algorithm can be used with an ESPAR antenna to estimate the directions of arrival of impinging signals with the same efficiency as in conventional cases. So it can be applied to the WACNet to improve the performance of the existing adaptive algorithms.

For the first time, we found a way to compute a correlation matrix in the reactance domain for the single-port output ESPAR antenna. Thus, many investigations can be done to adapt the conventional methods using a cross correlation matrix to the ESPAR antenna.

During my internship I have submitted five papers for conferences. Three papers called "Interference Cancellation in OFDM Receiver with Adaptive ESPAR Antenna" have been presented and published in IEICE national conference, IEICE RCS national conference and IEEE WCNC international conference. In the same way, two papers called "Reactance-Domain MUSIC Algorithm for ESPAR Antennas" have been presented and published in IEICE RCS national conference and IEEE WCNC international conference.



# Bibliography

- [1] K. Gyoda, Y. Ohna, Y. Kado and T. Ohira, "WACNet: Wireless Ad-Hoc Community Network," IEEE Inter. Conf. on Industrial Electronics, Control and Instrumentation, pp. 1153–1158, October 2000, Nagoya Japan.
- [2] The official bluetooth website, <http://bluetooth.com/>.
- [3] 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, pp. 1790–1800, July 2001.
- [4] D. V. Thiel and S. Smith, "Switched Parasitic Antennas for Cellular Communications," Artech House Publishers, ISBN: 1-58053-154-7.
- [5] E. Taillefer, "Implementation of adaptive controller for ESPAR antenna by using DSP and experiments for performance verification," Advanced Telecommunications Research Laboratories, 2001.
- [6] W. Y. Zou and Y. Wu, "COFDM: an overview," IEEE Trans. Broad., vol. 41, no. 1, pp. 1–8, March 1995.
- [7] A. Pandharipande, "Principles of OFDM," IEEE Potentials, pp. 16–19, April-May 2002.
- [8] K.-K. Wong, R. S.-K. Cheng, K. B. Letaief and R. D. Murch, "Adaptive Antennas at the Mobile and Base Stations in an OFDM/TDMA System," IEEE Trans. Commun., vol. 49, no. 1, pp. 195–206, January 2001.
- [9] M. Okada and S. Komaki "Pre-DFT combining space diversity assisted COFDM," IEEE Trans. VT., vol. 50, no. 2, pp. 487–496, March 2001.
- [10] R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," IEEE Trans. AP., vol. AP-34, no. 3, pp. 276–280, March 1986.
- [11] T. Ohira and K. Gyoda, "Hand-Held Microwave Direction-of-Arrival Finder Based on Varactor-Tuned Analog Aerial Beamforming," Proc. 2001 Asia-Pacific Microwave Conf., pp.585-588, Taipei, Dec. 2001.

- [12] M. Wax and J. Sheinvald, "Direction Finding of Coherent Signals via Spatial Smoothing for Uniform Circular Arrays," IEEE Trans. AP., vol. 42, no. 5, pp. 613-620, March 1994.
- [13] R. Martin-Cuerdo, D. Segovia-Vargas and M. Sierra-Perez, "Minimization of mutual coupling effects in eigenstructure algorithms based on modal analysis," IEEE ,1999.
- [14] J. G. Proakis and M. Salehi, "Contemporary Communication Systems using MATLAB," BookWare Companion Series, ISBN: 0-534-37173-6.
- [15] S. Haykin, "Adaptive Filter Theory," Second edition, Prentice Hall.
- [16] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing Principles, Algorithms, and Applications," Third edition, Prentice Hall International Editions, ISBN: 0-13-394289-9.
- [17] J. G. Proakis, "Digital Communications," Mc Graw Hill, ISBN: 0-07-050927-1.
- [18] J. Litva and T. K.-Y. Lo, "Digital Beamforming in Wireless Communications," Artech House Publishers, ISBN: 0-89006-712-0.

# APPENDIX A : MATLAB SOURCE CODES FOR ESPAR + OFDM

## OFDM.m

program name: OFDM.m comments: Interference Cancellation in OFDM Receiver with Adaptive ESPAR Antenna
--

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                                                                    %  
%                               ESPAR + OFDM                          %  
%                                                                    %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%-----  
%   Clear Workspace  
%-----  
  
clear all;  
  
fclose('all');  
  
%-----  
% Initialisation of global variables  
%-----  
  
disp(' ')  
Eb_No_dB = 30%input('Signal to Noise Ratio (dB) : ');  
disp(' ')  
  
N_pilot = 150%input('Number of step for the pilot sequence : ');  
disp(' ')  
  
mu = 550%input('Adaptation step : ');  
disp(' ')
```

```

delta = 5%input('Perturbation size of reactances : ');
disp(' ')

SOI_DoA = 0%input('DoA of SOI : ');
disp(' ')

delays = [5 17 21]%input('Delays of multipath signals : ');
disp(' ')

magnitudes = [0.5 0.5 0.5]%input('Magnitude of multipath signals : ');
disp(' ')

DoAs = [60 170 280]%input('DoAs of delayed signals : ');%
disp(' ')

disp(' ')
disp('press any key to continue')
disp(' ')
pause

% ESPAR antenna parameters
L=7                % size of ESPAR antenna

Zrange=[-300 300]  % range for the reactances in Ohm

% OFDM parameters
Nsub_Channel = 2^6 % number of subchannels

Guard_interval = 16 % Guard Interval

% Data sequence
write_out = 1;      % 1 to write In/Out files

P = 2^6             % length of data sequences

Nsymbol_mod = N_pilot*P*L % number of modulated symbols for the pilot sequence

Nsymbol_OFDM = Nsymbol_mod/Nsub_Channel % number of OFDM symbols

N_OFDM = Nsymbol_OFDM*(Nsub_Channel+Guard_interval) % length of OFDM data

file_in='binary_in.dat';

file_out='binary_out.dat';

eval(['!del ',file_out]);

%-----
% Channel impulse response : h(t) --> H(f)

```

```

%-----
% number of multipath signals
Ninterf=length(delays);

% We assume that the delayed path are no longer than the number of sub channels

Channel_impulse_response = zeros(Nsub_Channel,1);

Channel_impulse_response(1) = 1;

for counter = 1:Ninterf

    if delays(counter) <= Guard_interval
        Channel_impulse_response(delays(counter)+1) = magnitudes(counter);
    end

end

Channel_response_f = fft(Channel_impulse_response);

%-----
% Transmitter for the pilot sequence :
%     Binary Data -> Encode NRZ -> 4-MAQ Modulation -> OFDM
%-----

seq_pilot=pilot(N_pilot,P,L,write_out,file_in);

figure(1);
subplot(2,2,1)
plot(seq_pilot,'x')
xlim([-2 2])
ylim([-2 2])
title('Transmitted signals : 4-QAM constellation');

OFDM_signal=[];

for iteration = 1:Nsymbol_OFDM

    symbol = ifft(seq_pilot((iteration-1)*Nsub_Channel + 1 : iteration*Nsub_Channel));

    OFDM_symbol = [symbol(Nsub_Channel-Guard_interval + 1 : Nsub_Channel).',symbol.'].';

    OFDM_signal = [OFDM_signal.',OFDM_symbol.'].';

end

P_signal_OFDM=mean(abs(OFDM_signal).^2);

```

```

%-----
% Transmission Channel --> multipath + awgn
%-----

% add Additive White Gaussian Noise

gaussian_noise = randn(N_OFDM,1) + i * randn(N_OFDM,1);

Eb_No = 10^(Eb_No_dB / 10.)

P_bruit = P_signal_OFDM / (2*Eb_No);

OFDM_signal_plus_AWGN = OFDM_signal + sqrt(P_bruit)*gaussian_noise;

% SOI impinging on antenna
received_signals = OFDM_signal_plus_AWGN*steer_vector(SOI_DoA,L).';

% add impinging multipath signals on antenna to the SOI signal
if Ninterf~=0
    for n_interf=1:Ninterf

        received_signals = received_signals + magnitudes(n_interf)*...
            [zeros(delays(n_interf),1).',OFDM_signal_plus_AWGN(1:N_OFDM-...
                delays(n_interf)).'].*steer_vector(DoAs(n_interf),L).';

    end
end

%-----
% Remove Guard Interval -> FFT -> Equalization + Spatial Filtering
%-----

% Initialisation of the reactance vector
x0 = -j*50; % x0 is always set to -j50
x = zeros(L,1);
x(1) = x0;

deltax = ones(L,1)*delta;
deltax(1) = 0;

% compute of the current vector
i_current = reactances2current(x);

received_signals_rm_guard = [];
for iteration = 1:N_pilot

    block_ref = seq_pilot((iteration-1)*Nsub_Channel*L + 1 : iteration*Nsub_Channel*L);

    temp_block = received_signals((iteration-1)*(Nsub_Channel+Guard_interval)*...

```



```

L + 1 : iteration*(Nsub_Channel+Guard_interval)*L,:);

block_received_signals_rm_guard = [];
for sensor = 1:L

    block_received_signals_rm_guard = [block_received_signals_rm_guard.',...
        temp_block(sensor*(Guard_interval+Nsub_Channel)-Nsub_Channel + 1 :sensor*...
            (Nsub_Channel+Guard_interval),:).'].'];

end

received_signals_rm_guard =...
    [received_signals_rm_guard.',block_received_signals_rm_guard.'].'];

% main loop steepest gradient algorithm

block_PxL = block_received_signals_rm_guard;      % data block at iteration n
r = block_ref;                                   % reference signal at iteration n

% measure y(n)
estimation = block_PxL(1:P,:)*i_current;

%-----
temp = i_current.*steer_vector(SOI_DoA,L);
gain_SOI_DoA = abs(temp);
phase_SOI_DoA = angle(temp);
estimation = (1/gain_SOI_DoA)*estimation*exp(-i*phase_SOI_DoA);
%-----
%-----
for cpt = 1:Ninterf

    gain = abs(i_current.*steer_vector(DoAs(cpt),L))/gain_SOI_DoA;

    if delays(cpt) <= Guard_interval
        Channel_impulse_response(delays(cpt)+1) = magnitudes(cpt)*gain;
    end

end

%-----
Channel_response_f = fft(Channel_impulse_response);
%-----
Coeff_taps = Channel_response_f.' ./ (abs(Channel_response_f).^2+...
    P_bruit/P_signal_OFDM);
%-----

y = fft(estimation);

y_equalize = y.*Coeff_taps;      % Equalization assuming we know

```

```

                                % the Channel Impulse Response perfectly

y_decided = sign(real(y_equalize))+i*sign(imag(y_equalize));

cross_correlation(iteration) = abs(r(1:P)*y_equalize)/(sqrt(r(1:P)*...
    r(1:P))*sqrt(y_equalize*y_equalize));

rho0 = cross_correlation(iteration)*ones(1,L);
rho0(1) = 0.0;

for index=2:L

    % Xm=Xm+deltaXm
    x(index) = x(index) + deltax(index);

    if x(index) > Zrange(2)
        x(index) = Zrange(2);
    elseif x(index) < Zrange(1)
        x(index) = Zrange(1);
    end

    i_currentI = reactances2current(x);

    tmpI = block_PxL((index-1)*P+1:index*P,:);
    rI = r((index-1)*P+1:index*P,:);

    estimationI = tmpI*i_currentI;

    %-----
    temp = i_currentI.*steer_vector(SOI_DoA,L);
    gain_SOI_DoA = abs(temp);
    phase_SOI_DoA = angle(temp);
    estimationI = (1/gain_SOI_DoA)*estimationI*exp(-i*phase_SOI_DoA);
    %-----
    %-----
    for cpt = 1:Ninterf

        gain = abs(i_currentI.*steer_vector(DoAs(cpt),L))/gain_SOI_DoA;

        if delays(cpt) <= Guard_interval
            Channel_impulse_response(delays(cpt)+1) = magnitudes(cpt)*gain;
        end

    end

end

%-----
Channel_response_f = fft(Channel_impulse_response);
%-----
Coeff_taps = Channel_response_f.' ./ (abs(Channel_response_f).^2+...
    P_bruit/P_signal_OFDM);

```

```

%-----

yI = fft(estimationI);

yI_equalize = yI.*Coeff_taps;      % Equalization assuming we know
                                   % the Channel Impulse Response perfectly

yI_decided = sign(real(yI_equalize))+i*sign(imag(yI_equalize));

rho(index) = abs(rI'*yI_equalize)/(sqrt(rI'*rI)*sqrt(yI_equalize'*yI_equalize));

x(index) = x(index) - deltax(index);

end

x = x + mu*(rho-rho0).';

x(1) = x0;

for index=2:L
    if x(index) > Zrange(2)
        x(index) = Zrange(2);
    elseif x(index) < Zrange(1)
        x(index) = Zrange(1);
    end
end

i_current = reactances2current(x);

end

estimate_signal = received_signals_rm_guard * i_current;

%-----
temp = i_current.'*steer_vector(SOI_DoA,L);
gain_SOI_DoA = abs(temp);
phase_SOI_DoA = angle(temp);
estimate_signal = (1/gain_SOI_DoA)*estimate_signal*exp(-i*phase_SOI_DoA);
%-----
%-----
for cpt = 1:Ninterf

    gain = abs(i_current.'*steer_vector(DoAs(cpt),L))/gain_SOI_DoA;

    if delays(cpt) <= Guard_interval
        Channel_impulse_response(delays(cpt)+1) = magnitudes(cpt)*gain;
    end
end

```

```

end
%-----
Channel_response_f = fft(Channel_impulse_response);
%-----
Coeff_taps = Channel_response_f' ./ (abs(Channel_response_f).^2+...
    P_bruit/P_signal_OFDM);
%-----

received_baseband_signal = [];
for iteration = 1:Nsymbol_OFDM

    symbol = fft(estimate_signal((iteration-1)*...
        Nsub_Channel + 1 : iteration*Nsub_Channel));

    symbol_equalize = symbol.*Coeff_taps;      % Equalization assuming we know
                                              % the Channel Impulse Response perfectly

    received_baseband_signal = [received_baseband_signal.',symbol_equalize.'].';
end

%-----
% Preparation for the visualization
%-----

minTheta = 0;
maxTheta = 360;
deg_theta = minTheta:maxTheta-1;
theta = deg2rad(deg_theta);

all_direction_steer_vector = ones(1,maxTheta-minTheta);
for sensor=2:L
    all_direction_steer_vector=[all_direction_steer_vector;exp(i*(pi/2)*...
        cos(theta-(sensor-2)*(2*pi/(L-1))))];
end

temp_visu = i_current.'*all_direction_steer_vector;

gain = 20*log10(abs(temp_visu));

phase = angle(temp_visu);

%-----
% Receiver : 4-MAQ Modulation -> Encode NRZ -> Binary Data
%-----

subplot(2,2,4)

```

```

plot( received_baseband_signal,'x')
xlim([-2 2])
ylim([-2 2])
title('Receiver - Pilot sequence : 4-QAM constellation');

[r_binary_data]=receiver(received_baseband_signal,write_out,file_out);

%-----
% BER
%-----

calculation_BER(file_in,file_out);

%-----
% Graphs and Curves
%-----

subplot(2,2,2);
plot(deg_theta,gain,'',DoAs,gain(DoAs-(minTheta-1)),'r. ');
hold on
plot(SOI_DoA,gain(SOI_DoA-(minTheta-1)),'g. ');
hold off
title(['SOI : ',num2str(SOI_DoA),'          DOA : ',num2str(DoAs)]);
ylabel('ESPAR Response - Gain (dB)');

subplot(2,2,3);
polar_ploth(theta,gain,-25,2,'m-',5);
title(['SOI : ',num2str(SOI_DoA),'          DOA : ',num2str(DoAs)]);

figure(2)
step = 1:N_pilot;
subplot(2,2,1);
plot(step,cross_correlation)
title('Cross Correlation Coefficient');

subplot(2,2,3);
plot(deg_theta,phase,'',DoAs,phase(DoAs-(minTheta-1)),'r. ');
hold on
plot(SOI_DoA,phase(SOI_DoA-(minTheta-1)),'g. ');
hold off
title(['SOI : ',num2str(SOI_DoA),'          DOA : ',num2str(DoAs)]);
ylabel('ESPAR Response - Phase (radian)');

subplot(2,2,4);
polar_ploth(theta,phase,min(phase),2,'m-', max(phase));
title(['SOI : ',num2str(SOI_DoA),'          DOA : ',num2str(DoAs)]);

```

## pilot.m

<p>program name: pilot.m comments: Pilot sequence</p>
---

```
function seq_pilot=pilot(N_pilot,P,L,write_out,file_in)

% prepare for the pilot sequence

binary_data = [];
for n=1:N_pilot
    binary_data_block = randint(2*P,1);
    for sensor=1:L
        binary_data = [binary_data; binary_data_block];
    end
end

phase_NRZ = 1 - 2*binary_data(1:2:N_pilot*2*P*L);
quadrature_NRZ = 1 - 2*binary_data(2:2:N_pilot*2*P*L);
seq_pilot = phase_NRZ + i*quadrature_NRZ;

if (write_out)
    fid=fopen(file_in,'w');
    fwrite(fid,binary_data,'short');
    fclose(fid);
end;
```

## steer\_vector.m

<pre>program name: steer_vector.m comments: Steer vector for ESPAR antenna</pre>
--

```
% Forming a steer vector with respect to the DOA theta in radian

function steervector=steer_vector(deg_theta,L)

theta=deg2rad(deg_theta);

vector(1)=1.0;

for sensor=2:L

    vector(sensor)=exp(i*(pi/2)*cos(theta-(sensor-2)*(2*pi/(L-1))));

end

steervector = vector.';
```

## deg2rad.m

<pre>program name: deg2rad.m comments: Degree to radian</pre>
---

```
% convert degree angle to radian angle
```

```
function ret=deg2rad(in_deg)
```

```
if in_deg >=0
```

```
    ret = 2*pi*(in_deg/(360));
```

```
else
```

```
    ret = 2*pi*((in_deg+360)/360);
```

```
end
```



## reactances2current.m

<pre>program name: reactances2current.m comments: Reactances to current with admittance matrix</pre>
--

```
function cur=reactances2current(rea)

% Given reactance vector, calculate the current vector

VS=270; %arbitrarily

L=7;

% set the admittances
% experimental parameters

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

% set the 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];

% set the admittance vector
y0 = [y00 y10 y10 y10 y10 y10 y10].';

% calculate the current vector
cur=VS*inv(eye(L)+Y*diag(j*rea))*y0;
```

## receiver.m

<pre>program name: receiver.m comments: Receiver</pre>
--

```
% 4-MAQ Modulation -> Encode NRZ -> Binary Data

function [r_binary_data]=receiver(r_baseband_signal,write_out,file_out)

N = length(r_baseband_signal);

decided_symbol = sign(real(r_baseband_signal))+i*sign(imag(r_baseband_signal));

demodulated_symbol = (1.+i-(decided_symbol))*0.5;

temp(1:2:2*N) = real(demodulated_symbol);
temp(2:2:2*N) = imag(demodulated_symbol);

r_binary_data = temp.';

if (write_out)
    fid=fopen(file_out,'w');
    fwrite(fid,r_binary_data,'short');
    fclose(fid);
end;
```

## calculation\_BER.m

<pre>program name: calculation_BER.m comments: BER calculation</pre>
--

```
% Calculation of the Binary Error Rate
```

```
function BER=calculation_BER(file_in,file_out);
```

```
fid1=fopen(file_out,'r');
data_out=fread(fid1,inf,'short');
fclose(fid1);
```

```
fid1=fopen(file_in,'r');
data_in=fread(fid1,inf,'short');
fclose(fid1);
```

```
N=length(data_in);
```

```
BER = sum(xor(data_out(1:N),data_in(1:N)))/N;
```

```
S=sprintf('-----\n');
disp(S);
S=sprintf('\n      Binary Error Rate : %0.5g  \n',BER);
disp(S);
S=sprintf('-----\n');
disp(S);
```

## polar\_ploth.m

```
program name: polar_ploth.m
comments: Polar ploth
```

```
function polar_ploth(theta, r, centre_value, flag, S, max_mag)
%POLAR_PLOT    polar plot of power values with centre value defined in
%              typical H field orientation
%
%polar_plot(THETA, R, C, FIELD, flag, S, MAX_MAG) will plot the power values
%defined by their angle THETA and magnitude R in a polar form around a centre value of
%C. S is the line format you wish to use and is the same as the formats for the
%standard plot function
%
%MAX_MAG is the maximum magnitude of r that you want to set the axis to.
%e.g. if you are plotting multile plots you will want to set the maximum r axis
%to the maximum of all plots. If only plotting one plot then simply enter
%
%              polar_ploth(THETA, R, CENTRE_VALUE, FLAG, S, max(r))
%
%FLAG = 0 - Data plotted on new figure and polar axes drawn
%FLAG = 1 - Data plotted on new figure without polar axes
%FLAG = 2 - Data plotted on current figure and polar axes added
%FLAG = 3 - Data plotted on current figure without adding polar axes
%
%Also, any modifications to the axis plotting/representation can be adjusted in the first
%few lines of the polar_ploth function m file. The modifications are: Number of rings,
%the angle were the dB markings are displayed, the number of angle marks (radial lines)
%and axis color.
%
%The FLAG is useful for plotting multiple plots if you want to use a legend. For example,
%to plot 3 plots with a legend of the graphs use the following FLAG order: 1, 3, 2.
%Make sure the axis are plotted in the last trace so they will not appear in the 'legend'

%some setup parameters
number_of_rings = 4;    %number of rings shown in plot
dB_mark_angle = 45;    %angle where the dB markings are
no_angle_marks = 12;   %12 gives 30 degree spacing
line_color = 'k';      %color of axis - k is black

%creat figure if necessary
if flag == 0 | flag == 1
    figure
end

hold on
```

```

%make sure input is not bogus
if max_mag < centre_value
    text(-0.7*max_mag,0.1,'Centre value is larger than maximum input value');
    break;
end

if max_mag == centre_value
    text(-0.7*max_mag,0.1,'Centre value is same as maximum input value');
    break;
end

%nomalize max_mag to centre value
max_mag = max_mag - centre_value;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%normalize the r input value. Shift all values up by the centre value
%amount. If the CV is negative, then all values will be shifted up,
%but if the CV is positive, then all values will be shifted down.
r = r - centre_value;
for i = 1:length(r)
    if r(i) < 0
        r(i) = 0;
    end
    if r(i) > max_mag
        r(i) = max_mag;
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot the values
x = r.*cos(theta);
y = r.*sin(theta);
plot(x,y,S);

%if the flag is set to plotting on current figure, then quit out
%without drawing axis and values
if flag == 1 | flag == 3
    hold off;
    break;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Draw the polar axis

%Draw the concentric circles
phi = 0:0.01:2*pi;
rings = max_mag;
for i = 1:number_of_rings

```

```

    rings = [rings (number_of_rings-i)*max_mag/number_of_rings];
end

for i = 1:1:length(rings)
    x = rings(i)*cos(phi);
    y = rings(i)*sin(phi);
    plot(x,y, strcat(':',line_color))
end

%Draw the angle markers
phi = (0:360/no_angle_marks:359)*pi/180;
for i = 1:1:length(phi)
    x = max_mag*cos(phi(i));
    y = max_mag*sin(phi(i));
    plot([0 x],[0 y],strcat(':',line_color))
end

%Draw on text labeling the angles
phi = 0:360/no_angle_marks:359;
for i = 1:1:length(phi)
    x = max_mag*1.15*cos(phi(i)*pi/180);
    y = max_mag*1.15*sin(phi(i)*pi/180);
    text(x,y,num2str(phi(i)))
end
axis([-max_mag*1.2 max_mag*1.2 -max_mag*1.2 max_mag*1.2])

%Draw on the text labeling the magnitude
angle_of_mark = dB_mark_angle*pi/180;
for i = 1:1:length(rings)
    x = (rings(i)+0.05*max_mag)*cos(angle_of_mark);
    y = (rings(i)+0.05*max_mag)*sin(angle_of_mark);
    text(x,y,num2str(rings(i)+centre_value))
end

axis off

hold off

```

## APPENDIX B : MATLAB SOURCE CODES FOR AAA + MUSIC

### create\_matrix\_aaa\_line.m

<pre>program name: create_matrix_aaa_line.m comments: Creation of the cross correlation matrix for a linear AAA</pre>
---

```
% create_matrix.m
% Digital and Analog Modulated Signals
%
% Antenna: o o o o o o o
%
i=sqrt(-1); j=i;
Pd = 2000; % Pd: number of samples
Fd =1; % Fd: frequency
Fs = 4*Fd; % Fs: sample frequency
R = 0.5; % R: roll-off factor
Delay = 5; % Delay: for filtering [symbol]
No = 1; % No: power [W/Hz]
M=4; % M: number of modulation step: 0, 1, 2, 3

x1 = randint(Pd,1,M);
x2 = randint(Pd,1,M);
x3 = randint(Pd,1,M);
% data mapping
y1 = modmap(x1,Fd,Fs,'qask',M);
y2 = modmap(x2,Fd,Fs,'qask',M);
y3 = modmap(x3,Fd,Fs,'qask',M);
% filtering (raised cosine filter)
[rcv_a1,ti] = rcosflt(y1, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a2,ti] = rcosflt(y2, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a3,ti] = rcosflt(y3, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
% qam modulation
s1 = amodce(rcv_a1,Fs,'qam');
s2 = amodce(rcv_a2,Fs,'qam');
```

```

s3 = amodce(rcv_a3,Fs,'qam');

% (array_make.m)
% Generate Array received sigal
m=7; % m: number of sensors
p=3; % p: number of sources
fc=2e9; % carrier frequency(2GHz)
c = 3e8; % light speed
wavelen = c/fc; % wavelength
d = wavelen/2; % space between elements
ang = [ 10; -60; 20]; % DoAs
theta = ang(1:p);
nn = 1024; % nn: length of signals
sn1 = 5; % signal 1 SNR (dB)
sn2 = 5; % signal 2 SNR (dB)
sn3 = 5; % signal 3 SNR (dB)
SN = [ sn1; sn2; sn3];
sn = SN(1:p);
deg2rad = pi/180;
%
%
tt = 1:nn;
SS = [ s1(tt).'; s2(tt).'; s3(tt).'];
S = SS(1:p,:);
% noise
nr = randn(m,nn);
ni = randn(m,nn);
U = nr+j*ni;
%
%
Ps = S*S'/nn;
ps = diag(Ps);
refp= 2*10.^(sn/10);
tmp=sqrt( refp./ps );
S2 = diag(tmp)*S;
%
% Steering vectors matrix
%
A = exp(-j*2*pi*([0:m-1]'.*sin(theta'*deg2rad))*d/wavelen);
%
% observations
X = A*S2+U;
%
% cross correlation matrix
Rxx = X*X'/nn;
Rinv = inv(Rxx);
%
save array_data X A S2 U Rxx Rinv m nn wavelen d deg2rad theta p sn

```



## music\_aaa\_line.m

```
program name: music_aaa_line.m
comments: MUSIC algorithm for a linear AAA
```

```
% MUSIC DOA Estimator
% cmusic.m
%
% Antenna: o o o o o o
%
clear; clf
load array_data
%*****
pa=3; % number of interferences
% matrix decomposition
[Uxx, Sxx, Vxx] = svd(Rxx);
sv = diag(Sxx);
Us = Uxx(:,1:pa);
Un = Uxx(:,pa+1:m);
% steering vector
th=[-90:1:90]';
A= exp(-j*2*pi*([0:m-1]'*sin(th'*deg2rad))*d/wavelen)/sqrt(m);
% MUSIC Spectrum
num = 1;
En = Un'*A;
den =diag(En'*En);
doa = 10*log10(num./den);
% MUSIC Spectrum
ly = 45;
subplot(2,1,1);plot(th, doa)
hold on; quiver(theta(1:p), ly*ones(p,1), zeros(p,1), -sn,0);
hold off
axis([ min(th) max(th) -10 50])
title('MUSIC DOA Estimator')
xlabel('Angle [degree]')
ylabel('DOA Spectrum [dB]')
text(theta(1), ly,'s1')
text(theta(2), ly,'s2')
text(theta(3), ly,'s3')
grid
% Eigenvalues Distribution
subplot(2,1,2); plot(sv)
title('Distribution of Eigenvalues')
xlabel('Index')
ylabel('Value')
```



# APPENDIX C : MATLAB SOURCE CODES FOR AAA WITH MUTUAL COUPLING + MUSIC

create\_matrix\_aaa\_round\_mc.m

<p><b>program name:</b> create_matrix_aaa_round_mc.m <b>comments:</b> Creation of the cross correlation matrix for AAA with the same structure as the ESPAR antenna</p>
---

```
% create_matrix.m
% Digital and Analog Modulated Signals
%           o o
% Antenna :   o o o
%           o o
% + MUTUAL COUPLING
i=sqrt(-1); j=i;
Pd = 2000; % Pd: number of samples
Fd =1; % Fd: frequency
Fs = 4*Fd; % Fs: sample frequency
R = 0.5; % R: roll-off factor
Delay = 5; % Delay: for filtering [symbol]
No = 1; % No: power [W/Hz]
M=4; % M: number of modulation step: 0, 1, 2, 3

x1 = randint(Pd,1,M);
x2 = randint(Pd,1,M);
x3 = randint(Pd,1,M);
x4 = randint(Pd,1,M);
x5 = randint(Pd,1,M);
% data mapping
y1 = modmap(x1,Fd,Fs,'qask',M);
y2 = modmap(x2,Fd,Fs,'qask',M);
y3 = modmap(x3,Fd,Fs,'qask',M);
y4 = modmap(x4,Fd,Fs,'qask',M);
```

```

y5 = modmap(x5,Fd,Fs,'qask',M);
% filtering (raised cosine filter)
[rcv_a1,ti] = rcosflt(y1, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a2,ti] = rcosflt(y2, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a3,ti] = rcosflt(y3, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a4,ti] = rcosflt(y4, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a5,ti] = rcosflt(y5, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
% qam modulation
s1 = amodce(rcv_a1,Fs,'qam');
s2 = amodce(rcv_a2,Fs,'qam');
s3 = amodce(rcv_a3,Fs,'qam');
s4 = amodce(rcv_a4,Fs,'qam');
s5 = amodce(rcv_a5,Fs,'qam');
%
% Generate Array received signal
m=7; % m: number of sensors
p=4; % p: number of sources
fc=2e9; % carrier frequency(2GHz)
c = 3e8; % light speed
wavelen = c/fc; % wavelength
d = wavelen/2; % space between elements
ang = [ 60; 200; 240; 310]; % DoAs
theta = ang(1:p);
nn = 1024; % nn: length of signals
sn1 = 5; % signal 1 SNR (dB)
sn2 = 5; % signal 2 SNR (dB)
sn3 = 5; % signal 3 SNR (dB)
sn4 = 5; % signal 4 SNR (dB)
sn5 = 5; % signal 5 SNR (dB)
SN = [ sn1; sn2; sn3; sn4];
sn = SN(1:p);
deg2rad = pi/180;
%
%
tt = 1:nn;
SS = [ s1(tt).'; s2(tt).'; s3(tt).'; s4(tt).'];
S = SS(1:p,:);
% noise
nr = randn(m,nn);
ni = randn(m,nn);
U = nr+j*ni;
%
%
Ps = S*S'/nn;
ps = diag(Ps);
refp= 2*10.^(sn/10);
tmp=sqrt( refp./ps );
S2 = diag(tmp)*S;
%

```

```

% Steering vectors matrix
A=[];
for index=1 : p
    A(:,index) = steer_vector_roundmc(ang(index),m); % d=wavelen/2
end
Zs=500*eye(m);

% experimental parameters

y00 = 0.0008616-j*0.0120795;

y10 = -0.0006963+j*0.0036462;

y11 = 0.0044216-j*0.0071600;

y21 = 0.0009721+j*0.0047851;

y31 = -0.0005376-j*0.0011297;

y41 = 0.0001701-j*0.0002950;

% set the 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];

Z = inv(Y);

C = Zs*inv(Z+Zs);

% observations
X = C*A*S2+U;
%
% cross correlation matrix
Rxx = X*X'/nn;
Rinv = inv(Rxx);
%
save array_data X A S2 C U Rxx Rinv m nn wavelen d deg2rad theta p sn

```

## music\_aaa\_round\_mc.m

<pre>program name: music_aaa_round_mc.m comments: MUSIC algorithm for AAA with the same structure as the ESPAR antenna</pre>
--

```
% MUSIC DOA Estimator
% cmusic.m
%
%           o o
% Antenna :  o o o
%           o o
clear; clf
load array_data
%*****
pa=4; % number of interferences
% matrix decomposition
[Uxx, Sxx, Vxx] = svd(Rxx);
sv = diag(Sxx);
Us = Uxx(:,1:pa);
Un = Uxx(:,pa+1:m);
% steering vector
th=0:360;
minTheta = min(th);
maxTheta = max(th);
th_all = degree2radian(th);
A = ones(1,maxTheta-minTheta+1);
for sensor=2:m
    A=[A;exp(i*1*(pi/2)*cos(th_all-(sensor-2)*(2*pi/(m-1))))];
    % d=wavelen/2
end
A=A/sqrt(m);
% MUSIC Spectrum
num = 1;
En = Un'*C*A;
den =diag(En'*En);
doa = 10*log10(num./den);
% MUSIC Spectrum
ly = 55;
subplot(2,1,1);plot(th, doa)
hold on; quiver(theta(1:p), ly*ones(p,1), zeros(p,1), -sn,0);
hold off
axis([ min(th) max(th) 0 60])
title('MUSIC DOA Estimator')
xlabel('Angle [degree]')
ylabel('DOA Spectrum [dB]')
text(theta(1), ly,'s1')
text(theta(2), ly,'s2')
text(theta(3), ly,'s3')
```

```
text(theta(4), ly,'s4')
%text(theta(5), ly,'s5')
grid
% Eigenvalues Distribution
subplot(2,1,2); plot(sv)
title('Distribution of Eigenvalues')
xlabel('Index')
ylabel('Value')
```

## steer\_vector\_roundmc.m

<p>program name: steer_vector_roundmc.m comments: Steer vector for AAA with the same structure as the ESPAR antenna</p>
---

```
% Forming a steer vector with respect to the DOA theta in radian  
function steervector=steer_vector_roundmc(deg_theta,L)  
theta=degree2radian(deg_theta);  
vector(1)=1.0;  
for sensor=2:L  
    vector(sensor)=exp(i*1*(pi/2)*cos(theta-(sensor-2)*(2*pi/(L-1))));  
    % d=wavelen/2  
end  
steervector = vector.';
```



## degree2radian.m

<pre>program name: degree2radian.m comments: Degree to radian</pre>
---

```
% convert degree angle to radian angle
```

```
function ret=degree2radian(in_deg)
```

```
if in_deg >=0
```

```
    ret = 2*pi*(in_deg/(360));
```

```
else
```

```
    ret = 2*pi*((in_deg+360)/360);
```

```
end
```



## APPENDIX D : MATLAB SOURCE CODES FOR ESPAR + MUSIC

### create\_matrix\_espar.m

<p>program name: create_matrix_espar.m comments: Creation of the cross correlation matrix for ESPAR antenna</p>
---

```
% create_matrix.m
% Digital and Analog Modulated Signals
%           o o
% Antenna:  o o o
%           o o
i=sqrt(-1); j=i;
Pd = 2000; % Pd: number of samples
Fd = 1; % Fd: frequency
Fs = 4*Fd; % Fs: sample frequency
R = 0.5; % R: roll-off factor
Delay = 5; % Delay: for filtering [symbol]
No = 1; % No: power [W/Hz]
M=4; % M: number of modulation step: 0, 1, 2, 3

x1 = randint(Pd,1,M);
x2 = randint(Pd,1,M);
x3 = randint(Pd,1,M);
x4 = randint(Pd,1,M);
x5 = randint(Pd,1,M);
x6 = randint(Pd,1,M);
% data mapping
y1 = modmap(x1,Fd,Fs,'qask',M);
y2 = modmap(x2,Fd,Fs,'qask',M);
y3 = modmap(x3,Fd,Fs,'qask',M);
y4 = modmap(x4,Fd,Fs,'qask',M);
y5 = modmap(x5,Fd,Fs,'qask',M);
y6 = modmap(x6,Fd,Fs,'qask',M);
```

```

%filtering (raised cosine filter)
[rcv_a1,ti] = rcosflt(y1, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a2,ti] = rcosflt(y2, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a3,ti] = rcosflt(y3, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a4,ti] = rcosflt(y4, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a5,ti] = rcosflt(y5, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a6,ti] = rcosflt(y6, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
% qam modulation
s1 = amodce(rcv_a1,Fs,'qam');
s2 = amodce(rcv_a2,Fs,'qam');
s3 = amodce(rcv_a3,Fs,'qam');
s4 = amodce(rcv_a4,Fs,'qam');
s5 = amodce(rcv_a5,Fs,'qam');
s6 = amodce(rcv_a6,Fs,'qam');

% Generate Array received signal
L=7; % m: number of sensors
p=4; % p: number of sources
fc=2e9; % carrier frequency(2GHz)
c = 3e8; % light speed
wavelen = c/fc; % wavelength
ang = [ 30; 120; 230; 320]; % DoAs
theta = ang(1:p);
nn = 1000; % nn: length of signals
sn1 = 5; % signal 1 SNR (dB)
sn2 = 5; % signal 2 SNR (dB)
sn3 = 5; % signal 3 SNR (dB)
sn4 = 5; % signal 4 SNR (dB)
sn5 = 5; % signal 5 SNR (dB)
sn6 = 5; % signal 6 SNR (dB)
SN = [ sn1; sn2; sn3; sn4];
sn = SN(1:p);
deg2rad = pi/180;
%
%
tt = 1:nn;
SS = [ s1(tt).'; s2(tt).'; s3(tt).'; s4(tt).'];
S = SS(1:p,:);
% noise
nr = randn(L-1,nn);
ni = randn(L-1,nn);
U = nr+j*ni;
%
%
Ps = S*S'/nn;
ps = diag(Ps);
refp= 2*10.^(sn/10);
tmp=sqrt( refp./ps );

```

```

S2 = diag(tmp)*S;

#####

reactances_set = [1.9329,0.1868,1.6289,2.6231,2.6055,0.9120].'*1.0e+002;

received_signals = zeros(nn,L);
AA = [];
for cpt=1 : p
    received_signals = received_signals + S2(cpt,:).'*steer_vector_esparnew(theta(cpt),L).';
    AA = [AA,steer_vector_esparnew(theta(cpt),L)];
end

% Initialisation of the reactance vector
x0 = -j*50; % x0 is always set to -j50

I = [];

% Random set of reactances
x_fixed = reactances_set;

y=[];
for index=1 : L-1
    x = [x0,x_fixed.'].';

    % Compute of the current vector
    i_current = reactances2currentbis(x);
    I = [I;i_current.'];

    % Determination the ESPAR output
    y(:,index) = received_signals*i_current;

    % rotation of the reactances vector
    x_fixed = [x_fixed(L-1),x_fixed(1:L-2).'].';

end

#####

X = y. ';
X = X + U;

% cross correlation matrix
Rxx = X*X'/nn;
Rinv = inv(Rxx);
%
save array_data AA X S2 U I Rxx Rinv L nn wavelen deg2rad theta p sn

```

## music\_espar.m

<pre>program name: music_espar.m comments: Reactance-domain MUSIC algorithm for ESPAR antenna</pre>
---

```
% MUSIC DOA Estimator
% cmusic.m
%           o o
% Antenna : o o o
%           o o

clear; clf
load array_data
%*****
pa=4; % number of interferences
% matrix decomposition
[Uxx, Sxx, Vxx] = svd(Rxx);
sv = diag(Sxx);
Us = Uxx(:,1:pa);
Un = Uxx(:,pa+1:L-1);
% steering vector
th=0:0.1:360;
minTheta = min(th);
maxTheta = max(th);
th_all = degree2radian(th);
A = ones(1,(maxTheta-minTheta)*10+1);
for sensor=2:L
    A=[A;exp(i*(pi/2)*cos(th_all-(sensor-2)*(2*pi/(L-1))))];
end

% MUSIC Spectrum
num = 1;
En = Un'*I*A; %different from conventional cases
den =diag(En'*En);
doa = 10*log10(num./den);
% MUSIC Spectrum

ly = 45;
subplot(2,1,1);plot(th, doa)
hold on;
quiver(theta(1:p), ly*ones(p,1), zeros(p,1), -sn,0);

axis([ min(th) max(th) -10 50])
title('Reactance-domain MUSIC DoAs Estimator','FontSize',14)
xlabel('Angle [degree]','FontName','TimesNewRoman','FontSize',12)
ylabel('MUSIC Spectrum [dB]','FontName','TimesNewRoman','FontSize',12)
text(theta(1)+4, ly,'s1','FontSize',12)
text(theta(2)+4, ly,'s2','FontSize',12)
```

```

text(theta(3)+4, ly, 's3', 'FontSize', 12)
text(theta(4)+4, ly, 's4', 'FontSize', 12)
%text(theta(5), ly, 's5')
%text(theta(5), ly, 's5')
grid
% Eigenvalues Distribution
subplot(2,1,2); plot(sv)
title('Distribution of Eigenvalues')
xlabel('Index')
ylabel('Value')

% matrix of cross and auto powers
P=inv(AA'*I*I*AA)*(AA'*I)*(Rxx-Sxx(L-1,L-1)*eye(L-1))*(I*AA)*inv(AA'*I*I*AA)

```

## create\_matrix\_espar\_multipath.m

```
program name: create_matrix_espar_multipath.m
comments: Creation of the cross correlation matrix for ESPAR antenna in multipath
environment
```

```
% create_matrix.m
% Digital and Analog Modulated Signals
%           o o
% Antenna:  o o o
%           o o

i=sqrt(-1); j=i;
Pd = 2000; % Pd: number of samples
Fd = 1; % Fd: frequency
Fs = 4*Fd; % Fs: sample frequency
R = 0.5; % R: roll-off factor
Delay = 5; % Delay: for filtering [symbol]
No = 1; % No: power [W/Hz]
M=4; % M: number of modulation step: 0, 1, 2, 3

x1 = randint(Pd,1,M);

% data mapping
y1 = modmap(x1,Fd,Fs,'qask',M);

% filtering (raised cosine filter)
[rcv_a1,ti] = rcosflt(y1, Fd, Fs, 'fir/sqrt/Fs', R, Delay);

% qam modulation
s1 = amodce(rcv_a1,Fs,'qam');

delay=[3 7 11];

r = length(s1);
s2 = [s1(delay(1)+1:r).',zeros(delay(1),1).']';
s3 = [s1(delay(2)+1:r).',zeros(delay(2),1).']';
s4 = [s1(delay(3)+1:r).',zeros(delay(3),1).']';

% Generate Array received signal
L=7; % m: number of sensors
p=4; % p: number of sources
fc=2e9; % carrier frequency(2GHz)
c = 3e8; % light speed
wavelen = c/fc; % wavelength
```



```

ang = [50 120 230 320];      % DoAs
theta = ang(1:p);
nn = 1000;      % nn: length of signals
sn1 = 6; % signal 1 SNR (dB)
sn2 = 4; % signal 2 SNR (dB)
sn3 = 4; % signal 3 SNR (dB)
sn4 = 4; % signal 4 SNR (dB)
sn5 = 5; % signal 5 SNR (dB)
sn6 = 5; % signal 6 SNR (dB)
SN = [ sn1; sn2 ;sn3 ;sn4];
sn = SN(1:p);
deg2rad = pi/180;
%
%
tt = 1:nn;
SS = [ s1(tt).'; s2(tt).'; s3(tt).'; s4(tt).'];
S = SS(1:p,:);
% noise
nr = randn(L-1,nn);
ni = randn(L-1,nn);
U = nr+j*ni;
%
%
Ps = S*S'/nn;
ps = diag(Ps);
refp= 2*10.^(sn/10);
tmp=sqrt( refp./ps );
S2 = diag(tmp)*S;

#####

reactances_set = [1.9329,0.1868,1.6289,2.6231,2.6055,0.9120].'*1.0e+002;

received_signals = zeros(nn,L);

AA = [];
for cpt=1 : p
    received_signals = received_signals + S2(cpt,:).'*steer_vector_esparnew(theta(cpt),L).';
    AA = [AA,steer_vector_esparnew(theta(cpt),L)];
end

% Initialisation of the reactance vector
x0 = -j*50; % x0 is always set to -j50

I = [];

% Random set of reactances

```

```

x_fixed = reactances_set;

y=[];
for index=1 : L-1
    x = [x0,x_fixed.'].';

    % Compute of the current vector
    i_current = reactances2currentbis(x);
    I = [I;i_current.'];

    % Determination the ESPAR output
    y(:,index) = received_signals*i_current;

    % rotation of the reactances vector
    x_fixed = [x_fixed(L-1),x_fixed(1:L-2).'].';
end

#####

X = y. ';
X = X + U;

% cross correlation matrix
Rxx = X*X'/nn;
Rinv = inv(Rxx);
%
save array_data X S2 AA U I Rxx Rinv L nn wavelen deg2rad theta p sn

```

## music\_espar\_multipath.m

```
program name: music_espar_multipath.m
comments: Reactance-domain MUSIC algorithm for ESPAR antenna in multipath
environment
```

```
% MUSIC DOA Estimator
% cmusic.m
%           o o
% Antenna : o o o
%           o o

clear; clf
load array_data
%*****
pa=4; % number of interferences
% matrix decomposition
[Uxx, Sxx, Vxx] = svd(Rxx);
sv = diag(Sxx);
Us = Uxx(:,1:pa);
Un = Uxx(:,pa+1:L-1);
% steering vector
th=0:0.1:360;
minTheta = min(th);
maxTheta = max(th);
th_all = degree2radian(th);
A = ones(1,(maxTheta-minTheta)*10+1);
for sensor=2:L
    A=[A;exp(i*(pi/2)*cos(th_all-(sensor-2)*(2*pi/(L-1))))];
end
%A=A/sqrt(L);
% MUSIC Spectrum
num = 1;
En = Un'*I*A;
den =diag(En'*En);
doa = 10*log10(num./den);
% MUSIC Spectrum
ly = 45;
subplot(2,1,1);plot(th, doa)
hold on;
quiver(theta(1:p), ly*ones(p,1), zeros(p,1), -sn,0);

axis([ min(th) max(th) -10 50])
title('Reactance-domain MUSIC DoAs Estimator','FontSize',14)
xlabel('Angle [degree]','FontSize',14)
ylabel('MUSIC Spectrum [dB]','FontSize',14)
text(theta(1), ly,'s1')
text(theta(2), ly,'s2')
```

```

text(theta(3), ly,'s3')
text(theta(4), ly,'s4')
%text(theta(5), ly,'s5')
%text(theta(5), ly,'s5')
grid
% Eigenvalues Distribution
subplot(2,1,2); plot(sv)
title('Distribution of Eigenvalues')
xlabel('Index')
ylabel('Value')

% matrix of cross and auto powers
P=inv(AA'*I'*I*AA)*(AA'*I')*(Rxx-Sxx(L-1,L-1)*eye(L-1))*(I*AA)*inv(AA'*I'*I*AA)

```

## create\_matrix\_espar\_no\_periodic.m

```
program name: create_matrix_espar_no_periodic.m
comments: Creation of the cross correlation matrix for ESPAR antenna with no periodic
interferences
```

```
% create_matrix.m
% Digital and Analog Modulated Signals
%           o o
% Antenna:  o o o
%           o o

i=sqrt(-1); j=i;
Pd = 2000; % Pd: number of samples
Fd =1; % Fd: frequency
Fs = 4*Fd; % Fs: sample frequency
R = 0.5; % R: roll-off factor
Delay = 5; % Delay: for filtering [symbol]
No = 1; % No: power [W/Hz]
M=4; % M: number of modulation step: 0, 1, 2 ,3

x1 = randint(Pd,1,M);
x1 = x1*ones(1,6);
x2 = randint(Pd,6,M);
x3 = randint(Pd,6,M);
x4 = randint(Pd,6,M);
x5 = randint(Pd,6,M);
x6 = randint(Pd,6,M);
% data mapping
y1 = modmap(x1,Fd,Fs,'qask',M);
y2 = modmap(x2,Fd,Fs,'qask',M);
y3 = modmap(x3,Fd,Fs,'qask',M);
y4 = modmap(x4,Fd,Fs,'qask',M);
y5 = modmap(x5,Fd,Fs,'qask',M);
y6 = modmap(x6,Fd,Fs,'qask',M);
%filtering (raised cosine filter)
[rcv_a1,ti] = rcosflt(y1, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a2,ti] = rcosflt(y2, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a3,ti] = rcosflt(y3, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a4,ti] = rcosflt(y4, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a5,ti] = rcosflt(y5, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
[rcv_a6,ti] = rcosflt(y6, Fd, Fs, 'fir/sqrt/Fs', R, Delay);
% qam modulation
s1 = amodce(rcv_a1,Fs,'qam');
s2 = amodce(rcv_a2,Fs,'qam');
s3 = amodce(rcv_a3,Fs,'qam');
s4 = amodce(rcv_a4,Fs,'qam');
```

```

s5 = amodce(rcv_a5,Fs,'qam');
s6 = amodce(rcv_a6,Fs,'qam');

% Generate Array received signal
L=7; % m: number of sensors
p=4; % p: number of sources
fc=2e9; % carrier frequency(2GHz)
c = 3e8; % light speed
wavelen = c/fc; % wavelength
ang = [60;190; 250;300]; % DoAs
theta = ang(1:p);
nn = 2000; % nn: length of signals
sn1 =15; % signal 1 SNR (dB)
sn2 = 5; % signal 2 SNR (dB)
sn3 = 5; % signal 3 SNR (dB)
sn4 = 5; % signal 4 SNR (dB)
sn5 = 5; % signal 5 SNR (dB)
sn6 = 5; % signal 6 SNR (dB)
SN = [ sn1; sn2;sn3;sn4];
sn = SN(1:p);
deg2rad = pi/180;
%
%
tt = 1:nn;

#####

reactances_set = [1.9329,0.1868,1.6289,2.6231,2.6055,0.9120].'*1.0e+002;

% Initialisation of the reactance vector
x0 = -j*50; % x0 is always set to -j50

I = [];

% Random set of reactances
x_fixed = reactances_set;

y=[];
for index=1 : L-1

SS = [ s1(tt,index).'; s2(tt,index).';s3(tt,index).';s4(tt,index).'];
S = SS(1:p,:);
% noise
nr = randn(L-1,nn);
ni = randn(L-1,nn);

```

```

U = nr+j*ni;
%
%
Ps = S*S'/nn;
ps = diag(Ps);
refp= 2*10.^(sn/10);
tmp=sqrt( refp./ps );
S2 = diag(tmp)*S;
received_signals = zeros(nn,L);
for cpt=1 : p
    received_signals = received_signals + S2(cpt,:).'*steer_vector_esparnew(theta(cpt),L).';
end
x = [x0,x_fixed.'].';

% Compute of the current vector
i_current = reactances2currentbis(x);
I = [I;i_current.'];

% Determination the ESPAR output
y(:,index) = received_signals*i_current;

% rotation of the reactances vector
x_fixed = [x_fixed(L-1),x_fixed(1:L-2).'].';
end

#####

X = y.';
X = X + U;

% cross correlation matrix
Rxx = X*X'/nn;
Rinv = inv(Rxx);
%
save array_data X S2 U I Rxx Rinv L nn wavelen deg2rad theta p sn

```

## music\_espar\_no\_periodic.m

```
program name: music_espar_no_periodic.m
comments: Reactance-domain MUSIC algorithm for ESPAR antenna with no periodic
interferences
```

```
% MUSIC DOA Estimator
% cmusic.m
%           o o
% Antenna : o o o
%           o o

clear; clf
load array_data
%*****
pa=4; % number of interferences
% matrix decomposition
[Uxx, Sxx, Vxx] = svd(Rxx);
sv = diag(Sxx);
Us = Uxx(:,1:pa);
Un = Uxx(:,pa+1:L-1);
% steering vector
th=0:0.1:360;
minTheta = min(th);
maxTheta = max(th);
th_all = degree2radian(th);
A = ones(1,(maxTheta-minTheta)*10+1);
for sensor=2:L
    A=[A;exp(i*(pi/2)*cos(th_all-(sensor-2)*(2*pi/(L-1))))];
end
%A=A/sqrt(L);
% MUSIC Spectrum
num = 1;
En = Un'*I*A;
den =diag(En'*En);
doa = 10*log10(num./den);
% MUSIC Spectrum
ly = 45;
subplot(2,1,1);plot(th, doa)
hold on;
quiver(theta(1), ly, 0, -sn(1),0);
hold off
axis([ min(th) max(th) -20 40])
title('Reactance-domain MUSIC DoAs Estimator','FontSize',14)
xlabel('Angle [degree]','FontSize',14)
ylabel('MUSIC Spectrum [dB]','FontSize',14)
text(theta(1), ly-10,'s1')
%text(theta(2), ly,'s2')
```



```
%text(theta(3), ly,'s3')
%text(theta(4), ly,'s4')
%text(theta(5), ly,'s5')
%text(theta(5), ly,'s5')
grid
% Eigenvalues Distribution
subplot(2,1,2); plot(sv)
title('Distribution of Eigenvalues')
xlabel('Index')
ylabel('Value')
```

## steer\_vector\_esparnew.m

<pre>program name: steer_vector_esparnew.m comments: Steer vector for ESPAR antenna</pre>
---

```
% Forming a steer vector with respect to the DOA theta in radian

function steervector=steer_vector_esparnew(deg_theta,L)

theta=degree2radian(deg_theta);

vector(1)=1.0;

for sensor=2:L

    vector(sensor)=exp(i*(pi/2)*cos(theta-(sensor-2)*(2*pi/(L-1))));

end

steervector = vector.';
```

## degree2radian.m

<pre>program name: degree2radian.m comments: Degree to radian</pre>
---

```
% convert degree angle to radian angle

function ret=degree2radian(in_deg)

if in_deg >=0
    ret = 2*pi*(in_deg/(360));
else
    ret = 2*pi*((in_deg+360)/360);
end
```

## reactances2currentbis.m

<p><b>program name:</b> reactances2currentbis.m <b>comments:</b> Reactances to current with impedance matrix</p>
--

```
function cur=reactances2currentbis(rea)

% Given reactance vector, calculate the current vector

VS=300; %arbitrarily

L=7;

% set the admittances
% experimental parameters

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

% set the 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];

Z = inv(Y);

% set the admittance vector
u0 = [1 0 0 0 0 0 0].';

% calculate the current vector
cur=VS*inv(Z+diag(j*rea))*u0;
```