NTNU
Norwegian University of
Science and Technology

Sensor Array Signal Processing for Source Localization

Cristina Manzano García-Muñoz

Master of Science in Electronics

Submission date:	August 2008
Supervisor:	Hefeng Dong, IET
Co-supervisor:	Shefeng Yan, IET

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

Problem Description

Array signal processing has wide appplications in sonar, radar, wireless communications, radio astronomy, seismology, medical-imaging, speech acquisition, etc. Sources localization is one of the most important tasks in array signal processing and has been an active research area for many years. The main objective of this work is to study robust source localization methods for targets using an array of sensors. Direction-of-Arrival (DOA) estimation and range estimation are first studied, and then linked to source localization. The comparison of the existing source localization methods and a number of simulations should be included.

Assignment given: 15. January 2008
Supervisor: Hefeng Dong, IET

Sensor Array Signal Processing for Source Localization

Cristina Manzano García-Muñoz

August 2, 2008

Dedicated to my parents,
gracias por vuestro cariño y apoyo incondicionales.

Abstract

This work is a study about source localization methods, more precisely, about beamforming approaches. The necessary background theory is provided first, and then, further developed to explain the basis of each approach. The studied problem consists in an array of sensors in which the signal to process is impinging. Several examples of inciding signals are provided in order to compare the performance of the methods. The goal of the approaches is to find the Incident Signal Power and the Direction Of Arrival of the Signal (or Signals) Of Interest. With these information, the source can be located in angle and range. After the study, the conclusions will show which methods to chose depending on the application pursued. Finally, some ideas or guidelines about future investigation on the field, will be given.

Acknowledgements

All the ways we walked, all the obstacles we met, all the decisions we took, all the turns that detain us, all the times we got lost — and found again, all the dreams that guided us, the mistakes, the good choices, and the luck. At every step, people who came with us during a while — or forever, and some others who left us, leaving always something behind. Every path, every encounter, every, to arrive here, today, at this moment, in this place.

This work is the end of a long way, a way that started in UPV, in Spain, and ended some thousand kilometers further, in NTNU, in Norway. One year abroad, discovering lot of things I didn't know — or I had forgotten, learning so much about others — and about me, finding so many friends, sharing something difficult to explain… I have an indelible memory, and I know this year was one of the best years of my life. This work is an end, but is also a beginning. And I want to say, to all those who walked by my side, who made this possible, sincerely, thank you,

to my supervisors in NTNU, Hefeng Dong and Shefeng Yan,

to all my teachers in UPV,

to my friends,

to Milo,

to my parents, my sister, my family,…

With all my heart, my endless gratitude, once more, thank you.

Cristina Manzano García-Muñoz
Trondheim, August 2, 2008

Contents

List of Figures

List of Tables

Nomenclature

DAS	Delay and Sum
DCRCB	Doubly Constrained Robust Capon Beamformer
DL	Diagonal Loading
DOA	Direction Of Arrival
ISP	Incident Signal Power
NCCB	Norm Constrained Capon Beamformer
RCB	Robust Capon Beamformer
SCB	Standard Capon Beamformer
SOI	Signal Of Interest

Chapter 1

Motivation and Purpose

Array signal processing has wide applications in sonar, radar, wireless communications, radioastronomy, seismology, medical-imaging, speech acquisition, etc. Sources localization is one of the most important tasks in array signal processing and has been an active research area for many years.

The main objective of this work is to study robust source localization methods for targets using an array of sensors. Direction Of Arrival (angle) estimation and Incident Signal Power (range) estimation will be studied, and then linked to source localization.

The study will include a comparison of the existing source localization methods, also referred through the work as beamforming approaches, some solutions to improve their performance, and some simulations showing different cases of incident signals. The goal is to chose the most suitable method in each case and to explain the reason of this choice.

Chapter 2

Introduction to Beamforming

When an array of sensors receive a signal there is an implicit delay between the signal arriving at the different sensors because the signal has a finite velocity and the sensors are not located at the same location in space. This can be used by exploiting the fact that the delay among the sensors will be different depending on which direction the signal is coming from, and tuning the array to "look" in a specific direction. This process is know as beamforming, and can be used as a source localization method.

2.1 Beamformer Classification

Beamformers can be classified as either **data independent** or **statiscally optimum**, depending no how the weights are chosen. The weights in a data independent beamformer do not depend on the array data and are chosen to present a specified response for all signal/interference scenarios. The weights in a statiscally optimum beamformer are chosen based on the statistics of the array data to "optimize" the array response. In general the statiscally optimum beamformer places nulls in the directions of interfering sources in an attempt to maximize the signal to noise ratio at the beamformer output.

The statistics of the array data are not usually known and may change over time so **adaptive algorithms** are typically employed to determine the weights. The adaptive algorithm is designed so the beamformer response converges to a statiscally optimum solution. **Partially adaptive** beamformers reduce the adaptive algorithm computational load at the expense of a loss (designed to be small) in statistical optimality.

This work in centered in the statiscally adaptive techniques, that are most suitable for the demanded task of locating a source.

2.2 Basic Terminology and Concepts

The signal $s_0(t)$ arrives at the array of receivers (sensors). It is important to say, that one of them will be the *reference point*; this means that the delay to this sensor is considered to be null, and the signal is arriving to the rest of sensors with a delay relative to this one. The signal arrives with an angle θ_0, that is, the *Direction of Arrival (DOA)*.

Because of the DOA there is a delay in the signal arriving at each receiver of the array,

$$x_{0m}(t) = s_0(t - \tau_{0m}) + n(t) \tag{2.1}$$

or

$$X_{0m}(\omega) = e^{-j\omega\tau_{0m}} S_0(\omega) + N(\omega), \tag{2.2}$$

where the subscript m represents which receiver in the array; and $n(t)$, $N(\omega)$ in frequency domain, is the *noise*.

Then, an array of M elements can be characterized by

$$\begin{bmatrix} X_{01}(\omega) \\ X_{02}(\omega) \\ \vdots \\ X_{0M}(\omega) \end{bmatrix} = \begin{bmatrix} e^{-j\omega\tau_{01}} \\ e^{-j\omega\tau_{02}} \\ \vdots \\ e^{-j\omega\tau_{0M}} \end{bmatrix} S_0(\omega) + \mathbf{N}(\omega). \tag{2.3}$$

Equation (2.3) can be written as

$$\begin{bmatrix} X_{01}(\omega) \\ X_{02}(\omega) \\ \vdots \\ X_{0M}(\omega) \end{bmatrix} = \mathbf{A}(\theta_0) S_0(\omega) + \mathbf{N}(\omega), \tag{2.4}$$

where $\mathbf{A}(\theta_0)$ is the *array manifold vector* which is dependent on θ_0, that is, dependent on the delays that the signal suffers arriving at the different receivers. $\mathbf{A}(\theta_0)$ is also dependent on the frequency as it can be seen in (2.3).

In time domain, equation (2.4) is

$$\mathbf{x}_0(t) = \mathbf{a}(\theta_0) s_0(t) + \mathbf{n}(t), \tag{2.5}$$

but in fact, the sensors are not receiving only one signal but a combination of several ones, so,

$$\mathbf{x}(t) = \sum_{d=0}^{D} \mathbf{a}(\theta_d) s_d(t) + \mathbf{n}(t), \tag{2.6}$$

where D is the number of signals arriving to the array of receivers; $s_0(t)$ is the *Signal Of Interest (SOI)* and $s_1(t)$, $s_2(t)$, $\ldots$, $s_D(t)$ are the interfering ones.

2.2.1 Beamforming and Spatial Filtering

Depending on the nature of the signal (narrowband or broadband) it is possible to do a first distinction between beamformers [1].

When processing narrowband signals; the beamformer samples the propagating wave field in space. The output at time k, $y(k)$, is given by a linear combination of the data at the M sensors at time k

$$y(k) = \sum_{m=1}^{M} w_m^* x_m(k), \tag{2.7}$$

where * represents complex conjugate and w_m are the *weights* of the beamformer. It is conventional to multiply the data by conjugates of the weights to simplify notation. It is assumed throughout that the data and weights are complex since in many applications a quadrature receiver is used at each sensor to generate in phase and quadrature (I and Q) data. Each sensor is assumed to have any necessary receiver electronics and an A/D converter if beamforming is performed digitally.

When processing signals of significant frequency extent (broadband); the beamformer samples the propagating wave field in both space and time. The output in this case can be expressed as

$$y(k) = \sum_{m=1}^{M} \sum_{p=0}^{K-1} w_{m,p}^* x_m(k-p), \tag{2.8}$$

where $K-1$ is the number of delays in each of the M sensor channels. If the signal at each sensor is viewed as an input, then a beamformer represents a multi-input single out-put system. The remainder of the paper, however, will be focused on narrowband beamformers.

To simplify notation, (2.7) and (2.8) can be written as

$$y(k) = \mathbf{w}^H \mathbf{x}(k) \tag{2.9}$$

by appropiately defining a weight vector $\mathbf{w}$ and data vector $\mathbf{x}(k)$. The superscript H represents Hermitian (complex conjugate) transpose and vectors are assumed to be column vectors. Equation (2.9) can be written as

$$y = \mathbf{w}^H \mathbf{x}, \tag{2.10}$$

where the time index has been dropped to make notation easier.

2.2.2 Second Order Statistics

Second order statistics as power or variance play an important role in the evaluation of the beamformer performance. It will be assumed that the data received at the sensors is zero mean throughout the paper. In order to explain this section in a clear way, it will be assumed also that there is no noise or interferring signals arriving at the sensors, that means that **equation (2.6) will be**

$$\mathbf{x}(t) = \mathbf{a}(\theta)s(t)$$

just in this subsection.

The *variance* or *expected power* of the beamformer output is given by

$$E\{|y|^2\} = \mathbf{w}^H E\{\mathbf{x}\mathbf{x}^H\}\mathbf{w}. \tag{2.11}$$

If the data is wide sense stationary, the *data covariance matrix*,

$$\mathbf{R_x} = E\{\mathbf{x}\mathbf{x}^H\}, \tag{2.12}$$

is independent of time. The wide sense stationary assumption is used in developing statiscally optimal beamformers and in evaluating steady state performance.

Suppose $\mathbf{x}$ represents samples from a uniformly sampled time series having a power spectral density $S(\omega)$ and no energy outside of the spectral band $[\omega_a, \omega_b]$. $\mathbf{R_x}$ can be expressed in terms of the power spectral density of the data using the Fourier transform relationship as

$$\mathbf{R_x} = \frac{1}{2\pi}\int_{\omega_a}^{\omega_b} S(\omega)\mathbf{a}(\theta)\mathbf{a}^H(\theta)d\omega, \tag{2.13}$$

where $\mathbf{a}(\theta)$ is the already known array manifold vector that appeared in equation (2.5). As it has been said before, it is dependent on frequency, i.e. on ω.

When a source is narrowband of frequency ω_0, $\mathbf{R_x}$ can be represented as

$$\mathbf{R_x} = \sigma_s^2 \mathbf{a}(\theta)\mathbf{a}^H(\theta)\big|_{\omega=\omega_0}, \tag{2.14}$$

where σ_s^2 is the source variance or power, and the array manifold vector is evaluated at $\omega = \omega_0$.

The conditions under which a source can be considered narrowband depend on both the source bandwidth and the time over which the source is observed.

2.3 Snapshot Model

In order to characterize the arriving signal, not only, but several time samples will be computed, this is the *Snapshot Model* [2].

As said before, it is necessary to obtain second order statistics, more precisely, the *data covariance matrix.* However, in practical applications, the matrix $\mathbf{R_x}$ is unavailable and is replaced by the *sample covariance matrix* [3]

$$\tilde{\mathbf{R}}_x = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}(n)\mathbf{x}^H(n) = \frac{1}{N}\mathbf{X}\mathbf{X}^H, \tag{2.15}$$

where N is the number of snapshots available. When the number of snapshots is big enough, the sample covariance matrix is equivalent to the data covariance matrix, $\tilde{\mathbf{R}}_x \xrightarrow[N\to\infty]{} \mathbf{R_x}$.

Using equation (2.6) into (2.15), the data covariance matrix of the incoming signal formed by the SOI and the interfering ones can be estimated by

$$\mathbf{R_x} \approx \sum_{d=0}^{D} \sigma_d^2 \mathbf{a}(\theta_d)\mathbf{a}^H(\theta_d) + \mathbf{Q}, \tag{2.16}$$

where $\mathbf{Q}$ is the noise covariance matrix.

2.4 Traditional Approaches and State of the Art

In the simplest case of beaforming (**Delay and Sum**), the weights are

$$\mathbf{w} = \mathbf{a}(\theta). \tag{2.17}$$

Substituting this weights in (2.10), and using also (2.6),

$$y = \mathbf{w}^H \mathbf{x} = \mathbf{a}^H(\theta) \left(\sum_{d=0}^{D} \mathbf{a}(\theta_d) s_d + \mathbf{n} \right). \tag{2.18}$$

Comparing this result with the estimated data covariance matrix, (2.16), it can be seen a clear correspondance between the beamformer output y and the estimated data covariance matrix $\mathbf{R_x}$.

As it can be seen in the figure, by changing the angle θ in the weights, the beamformer output varies as well. This output will be maximum when $\theta = DOA$ of the SOI; in other words, this is a way of finding out the direction of arrival in which the source is emitting.

The estimated data covariance matrix can be used also to figure out the *Incident Signal Power (ISP)*, and with this the *range*, it is, the distance between the source and the sensors.

Although the idea is simple this beamformer is not very commonly used in practical applications because it do not work too well. The main disadvantage is that no nulls are placed directly in jamming signal locations. The **Delay and Sum beamformer** seeks only to enhance the signal in the direction to which the array is currently steered, and therefore, it is not possible to resolve power of two sources placed closer than a beamwidth.

In order to improve the above beamformer, researchers proposed the **Standard Capon Beamformer (SCB)**. The idea is try to minimize the power contributing by noise and any signals coming from other directions than the DOA of the SOI while mantaining a fixed gain in the direction of interest. This way the SCB has better resolution and much better interference rejection capability than the DAS beamformer.

The traditional techniques commented before are *ad hoc techniques*. In theory they are suppose to work well, but in practical applications the desired signal can be present during the beamformer training data, and also the assumptions on the nature of the desired signal and/or interference can be violated. In the case of the SCB, the problem arises when the array steering vector is not accurately known. This problems combined with the finite sample support, the inherent nonstationary nature of the underlying environmental processes, array manifold errors, etc. may lead to the need of a **robust adaptive beamforming** [4].

As commented before, one of the main problems that occur in practical adaptive array processing is the mismatch between the desired signal steering vector and the actual steering vector, that results in a *steering vector uncertainty.* Adaptive array techniques are known to be very sensitive to even slight mismatches of such a type that may occur due to signal pointing errors, imperfect array calibration, source local scattering, wavefront distortions, etc. All these effects result in suppression of the desired signal component. This phenomenon is commonly referred to as *signal self-nulling.*

There are several ad hoc approaches existing to overcome arbitrary desired signal mismatches, such as the **diagonal loading** of the sample covariance matrix, widely used for its simplicity.

However, how to select the loading level remains a crucial and open problem; the chosen loadings are not directly related to the steering vector uncertainty, so they are not guaranteed to be always optimal when the uncertainty changes [5].

One of the recent theoretically rigorous and powerful approaches to robust beamforming in the presence of an arbitrary unknown steering signal mismatch is based on **worst-case performance optimization**. These methods determine the optimal loading by defining the so-called *uncertainty set* [6, 7]. Adaptively choosing the loading according to the steering vector uncertainty, these approaches tend to outperform the ad hoc techniques previously mentioned. However, optimal loading is still solved mainly by iteration at present (by a Newton's method[1] for example). The iterative methods may suffer from slow convergence or nonconvergence unless the initial point for searching is selected very carefully. These slow convergence or non-convergence cases mean a heavy computational burden; and also, these iterative methods help little in revealing what factors can affect the optimal loading and how to affect it.

[1] More information about this method will be provided in Chapter 3 Approaches.

Since the essence of these **worst-case performance methods** is to improve robustness by imposing a diagonal matrix onto the covariance matrix, it is still useful and necessary to study how to determine the optimal loading according to such parameters as the steering vector uncertainty, the noise power, the source power, and so on. Recent advances on the field are focused on trying to obtain the optimal loading, and on finding out the parameters that affect this loading.

Chapter 3

Approaches

As seen in the introduction[1], several methods exist that implement beamforming. The goal of this chapter is to explain in detail these methods. For further information, the appendix includes the complete code of all programs used in this work. In the following chapter, several numerical examples will be given in order to compare the different approaches and the advantages and disadvantages of using each one.

3.1 Delay And Sum Beamformer

The underlying idea of Delay and Sum (DAS) beamforming is that when an electromagnetic signal impinges upon the aperture of the antenna array, the element outputs, added together with appropriate amounts of delays, reinforce signals with respect to noise or signals arriving at different directions. The delays required depend on the physical spacing between the elements in the array. The geometrical arrangement of elements and weights associated with each element are crucial factors in defining the array's characteristics.

As commented before, the weights in the DAS beamformer[2] are $\mathbf{w} = \mathbf{a}(\theta_0)$. This idea is exploded to obtain the ISP as

$$\mathbf{ISP} = \mathbf{a}(\theta_0)^H \tilde{\mathbf{R}}_x \mathbf{a}(\theta_0). \tag{3.1}$$

where $\tilde{\mathbf{R}}_x$ is the already seen sample covariance matrix.

[1] Chapter 2 Introduction to Beamforming.

[2] In this chapter, the subscript 0, as in $\mathbf{a}(\theta_0)$, references the SOI.

This is a very simple method but it seeks only to enhance the signal in the direction to which the array is currently steered, and so the limitations can be seen in the experimental results.

3.2 Standard Capon Beamformer

The Standard Capon Beamformer has better resolution and much better interference rejection capability than the data-independent beamformer when the array steering vector corresponding to the SOI is accurately known.

The undelying idea of this beamformer is try to minimize the power contributing by noise and any signals coming from other directions than the DOA of the SOI while mantaining a fixed gain in the direction of interest (this can be viewed as a *sharp spatial bandpass filter*) [8]. The optimal weights can be found using for example, the technique of Lagrange multipliers, resulting in

$$\mathbf{w} = \frac{\tilde{\mathbf{R}}_x^{-1}\mathbf{a}(\theta_0)}{\mathbf{a}^H(\theta_0)\tilde{\mathbf{R}}_x^{-1}\mathbf{a}(\theta_0)}, \tag{3.2}$$

and the power is therefore

$$\mathbf{ISP} = \frac{1}{\mathbf{a}^H(\theta_0)\tilde{\mathbf{R}}_x^{-1}\mathbf{a}(\theta_0)}. \tag{3.3}$$

Although this method works better than the former one, it lacks robutness in the presence of array steering vector errors.

3.3 Robust Capon Beamforming

Robust Capon Beamforming is a natural extension of the Standard Capon Beamforming to the case of uncertain steering vectors. This beamformer can no longer be expressed in a closed form, but it can be efficiently computed [9].

As it has been said, the problem can be stated as follows: extend the SCB so as to be able to accurately determine the power of the SOI even when only an imprecise knowledge of its steering vector, $\mathbf{a}(\theta_0)$, is available. In other words, it is assumed that the only knowledge about $\mathbf{a}(\theta_0)$ is that it belongs to the following uncertainty ellipsoid

$$[\mathbf{a}(\theta_0) - \bar{\mathbf{a}}]^* \, \mathbf{C}^{-1} \, [\mathbf{a}(\theta_0) - \bar{\mathbf{a}}] \leq 1 \tag{3.4}$$

where $\bar{\mathbf{a}}$ and $\mathbf{C}$ (a positive definite matrix) are given.

A particular case of equation (3.4) will be considered in the numerical examples. This consideration can be assumed when the array calibration errors are relatively small (and can be neglected) but the knowledge of the DOA is inaccurate, in other words, the DOA of the SOI is assumed to be $\theta_0 + \Delta$ instead of θ_0. In this case $\bar{\mathbf{a}}$ is set to $\bar{\mathbf{a}} = \mathbf{a}(\theta_0 + \Delta)$. If also $\mathbf{C}$ is choosen as $\mathbf{C} = \epsilon\mathbf{I}$, equation (3.4) becomes

$$\|\mathbf{a}(\theta_0) - \bar{\mathbf{a}}\|^2 \leq \epsilon \qquad \bar{\mathbf{a}} = \mathbf{a}(\theta_0 + \Delta) \tag{3.5}$$

To avoid ambiguities, it will be assumed also that

$$\|\mathbf{a}(\theta_0)\|^2 = M \tag{3.6}$$

and

$$\|\bar{\mathbf{a}}\|^2 = M \tag{3.7}$$

where $\|\cdot\|$ denotes the Euclidean norm and M the number of sensors in the array.

These assumptions are reasonable for many scenarios including the cases of the look direction error and phase perturbations. They are violated when the array response vector also has gain perturbations. However, if the gain perturbations are small, the norm constraint still holds approximately.

3.3.1 Diagonal Loading

Diagonal Loading[3] has been a popular approach to improve the robustness of the Standard Capon Beamformer. The Diagonal Loading approaches are derived by imposing an additional quadratic constraint either on the Euclidean norm of the weight vector itself or on its difference from a desired weight vector. In this work, the constraint is imposed on the difference, so using equation (3.5) it can be stated the following quadratic problem [6]:

$$\min_{\mathbf{a}} \mathbf{a}^* \mathbf{R}^{-1} \mathbf{a} \qquad \|\mathbf{a} - \bar{\mathbf{a}}\|^2 = \epsilon. \tag{3.8}$$

This problem can be solved by using the *Lagrange multiplier methodoly*, which is based on the function

$$f = \mathbf{a}^* \mathbf{R}^{-1} \mathbf{a} + \lambda \left(\|\mathbf{a} - \bar{\mathbf{a}}\|^2 - \epsilon \right) \tag{3.9}$$

[3] To simplify notation, in this subsection $\mathbf{a}(\theta_0)$ is $\mathbf{a}$, and $\tilde{\mathbf{R}}_x$ is $\mathbf{R}$.

where $\lambda \geq 0$ is the Lagrange multiplier [10].

From equation (3.9) it can be obtained

$$g(\lambda) = \sum_{m=1}^{M} \frac{|z_m|^2}{(1+\lambda\gamma_m)} = \epsilon. \tag{3.10}$$

In this equation z_m denote the mth element of $\mathbf{z}$, where $\mathbf{z} = \mathbf{U}^*\bar{\mathbf{a}}$ and $\mathbf{R} = \mathbf{U}\boldsymbol{\Gamma}\mathbf{U}^*$. The columns of $\mathbf{U}$ contain the eigenvectors of $\mathbf{R}$, and the diagonal elements of the diagonal matrix $\boldsymbol{\Gamma}$, $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_M$ are the corresponding eigenvalues.

The goal is to determine λ, it is, the quantity of Diagonal Loading. To do that, it is assumed that the solution $\lambda > 0$ is unique and that it belongs to the following interval:

$$\frac{\|\bar{\mathbf{a}}\| - \sqrt{\epsilon}}{\gamma_1\sqrt{\epsilon}} \leq \lambda \leq \min\left\{\left(\frac{1}{\epsilon}\sum_{m=1}^{M}\frac{|z_m|^2}{\gamma_m^2}\right)^{1/2}, \frac{\|\bar{\mathbf{a}}\| - \sqrt{\epsilon}}{\gamma_M\sqrt{\epsilon}}\right\}. \tag{3.11}$$

The problem can be solved by a Newton's method. From equation (3.10) it can be stated

$$f(\lambda) = g(\lambda) - \epsilon = 0. \tag{3.12}$$

Applying Newton's method to $f(\lambda)$ leads to

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)}. \tag{3.13}$$

where the subscript n represents the nth iteration. The value of λ can be obtained by iteration in a very simple way. The only difficulty is that it can be hard to find the correct starting value λ_0 that makes the method converge.

Once the Lagrange multiplier λ is determined, the weights can be obtained as

$$\mathbf{w} = \frac{(\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{a}}{\mathbf{a}^H(\mathbf{R} + \lambda\mathbf{I})^{-1}\mathbf{a}}, \tag{3.14}$$

and the power is

$$\mathbf{ISP} = \frac{1}{\bar{\mathbf{a}}^*\mathbf{U}\boldsymbol{\Gamma}\left(\lambda^{-2}\mathbf{I} + 2\lambda^{-1}\boldsymbol{\Gamma} + \boldsymbol{\Gamma}^2\right)^{-1}\mathbf{U}^*\bar{\mathbf{a}}}. \tag{3.15}$$

3.3.2 Norm Constrained Capon Beamformer

As it has been said before, SCB has better resolution and much better interference rejection capability than the DAS Beamformer, although it lacks robustness in the presence of array steering vector errors. Diagonal Loading method improves the robustness of the SCB, but it is not clear how to choose the diagonal loading level based on information about the uncertainty of the array steering vector.

Norm Constrained Capon Beamformer (NCCB) approach [7] uses a norm constraint on the weight vector to improve the robustness against array steering vector errors and noise, more precisely, it imposes an additional constraint on the Euclidean norm of $\mathbf{w}$. Consequently, the beamforming problem is formulated as follows:

$$\min_{\mathbf{w}} \mathbf{w}^*\mathbf{R}\mathbf{w} \quad \text{subject to} \quad \begin{aligned} \mathbf{w}^*\bar{\mathbf{a}} &= 1 \\ \|\mathbf{w}\|^2 &\leq \zeta. \end{aligned} \tag{3.16}$$

Let S be the set defined by the constraints in (3.16). In addition, let

$$g_1(\mathbf{w}, \lambda, \mu) = \mathbf{w}^*\mathbf{R}\mathbf{w} + \lambda\left(\|\mathbf{w}\|^2 - \zeta\right) + \mu\left(-\mathbf{w}^*\bar{\mathbf{a}} - \bar{\mathbf{a}}^*\mathbf{w} + 2\right) \tag{3.17}$$

where λ and μ are the *real-valued Lagrange multipliers* with μ being arbitrary and $\lambda \geq 0$ satisfying $\mathbf{R} + \lambda\mathbf{I} > 0$ so that $g_1(\mathbf{w}, \lambda, \mu)$ can be minimized with respect to $\mathbf{w}$. Then

$$g_1(\mathbf{w}, \lambda, \mu) \leq \mathbf{w}^*\mathbf{R}\mathbf{w} \quad \text{for any} \quad \mathbf{w} \in S \tag{3.18}$$

with equality on the boundary of S.

As in the former approach, also in this one it is necessary to solve a problem using a Newton's Method. The problem is the following:

$$\frac{\displaystyle\sum_{m=1}^{M} \frac{|z_m|^2}{(\gamma_m + \hat{\lambda})^2}}{\left[\displaystyle\sum_{m=1}^{M} \frac{|z_m|^2}{(\gamma_m + \hat{\lambda})}\right]^2} = \zeta \tag{3.19}$$

that must be solved only when the condition

$$\zeta < \frac{\bar{\mathbf{a}}^*\mathbf{R}^{-2}\bar{\mathbf{a}}}{\left[\bar{\mathbf{a}}^*\mathbf{R}^{-1}\bar{\mathbf{a}}\right]^2} \tag{3.20}$$

is satisfied.

As in DL, z_m denote the mth element of $\mathbf{z}$. Also, $\mathbf{z} = \mathbf{U}^*\bar{\mathbf{a}}$ and $\mathbf{R} = \mathbf{U}\mathbf{\Gamma}\mathbf{U}^*$. The columns of $\mathbf{U}$ contain the eigenvectors of $\mathbf{R}$, and the diagonal elements of the diagonal matrix $\mathbf{\Gamma}$, $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_M$ are the corresponding eigenvalues.

The value of $\hat{\lambda}$ is obtained from (3.19) by a Newton's Method[4] using also the knowledge that $\hat{\lambda}$ is upper bounded by

$$\hat{\lambda} \leq \frac{\gamma_1 - (M\zeta)^{1/2}\gamma_M}{(M\zeta)^{1/2} - 1} \tag{3.21}$$

and lower bounded by 0.

Then, the obtained $\hat{\lambda}$ is used in

$$\hat{\mathbf{w}} = \frac{\mathbf{U}(\mathbf{\Gamma} + \hat{\lambda}\mathbf{I})^{-1}\mathbf{U}^*\bar{\mathbf{a}}}{\bar{\mathbf{a}}^*\mathbf{U}(\mathbf{\Gamma} + \hat{\lambda}\mathbf{I})^{-1}\mathbf{U}^*\bar{\mathbf{a}}} \tag{3.22}$$

to obtain the weights of the beamformer.

The SOI power estimate of NCCB is

$$\mathbf{ISP} = \frac{\bar{\mathbf{a}}^*\mathbf{U}(\mathbf{\Gamma} + \hat{\lambda}\mathbf{I})^{-2}\mathbf{\Gamma}\mathbf{U}^*\bar{\mathbf{a}}}{\left[\bar{\mathbf{a}}^*\mathbf{U}(\mathbf{\Gamma} + \hat{\lambda}\mathbf{I})^{-1}\mathbf{U}^*\bar{\mathbf{a}}\right]^2} \tag{3.23}$$

which is easily obtained using $\mathbf{ISP} = \hat{\mathbf{w}}^*\mathbf{R}\hat{\mathbf{w}}$.

3.3.3 Doubly Constrained Robust Capon Beamformer

The problem with NCCB is that the choice of ζ is not easy to make. In particular, this choice is not directly linked to the ϵ in equation (3.5) or the uncertainty of the SOI steering vector. The Doubly Constraint Robust Capon Beamformer (DCRCB) algorithm does not suffer from this problem.

DCRCB is a natural extension of the SCB, which it has been obtained via covariance matrix fitting, to the case of uncertain steering vectors by enforcing a double constraint on the array steering vector, it means, a constant norm constraint, equation (3.6), and a spherical uncertainty set constraint,

[4] See (3.12) and (3.13).

equation (3.5). This way, it can be obtained a robust estimate of σ_0^2, where σ_0^2 is the ISP of the SOI, without any intermediate calculation of a vector $\mathbf{w}$ [7]:

$$\begin{aligned} \max_{\sigma^2,\mathbf{a}} \sigma^2 \quad \text{subject to} \quad & \mathbf{R} - \sigma^2 \mathbf{a}\mathbf{a}^* \geq 0 \\ & \|\mathbf{a} - \bar{\mathbf{a}}\|^2 \leq \epsilon \\ & \|\mathbf{a}\|^2 = M \end{aligned} \tag{3.24}$$

where $\bar{\mathbf{a}}$ is given and satisfies (3.7), and ϵ is also given and satisfies $\epsilon > 0$.

Using the fact that, for given $\mathbf{a}$, the solution of (3.24), σ^2, is obtained by $\sigma_0^2 = 1/\left(\mathbf{a}^*\mathbf{R}^{-1}\mathbf{a}\right)$, the DCRCB problem can be reduced to:

$$\begin{aligned} \min_{\mathbf{a}} \mathbf{a}^*\mathbf{R}^{-1}\mathbf{a} \quad \text{subject to} \quad & \|\mathbf{a} - \bar{\mathbf{a}}\|^2 \leq \epsilon \\ & \|\mathbf{a}\|^2 = M. \end{aligned} \tag{3.25}$$

Let $\hat{\mathbf{a}}$ denote the solution to the above optimization problem. The SOI power estimate is then calculated as

$$\hat{\sigma}_0^2 = \frac{1}{\hat{\mathbf{a}}^*\mathbf{R}^{-1}\hat{\mathbf{a}}}. \tag{3.26}$$

Using $\|\mathbf{a}\|^2 = \|\bar{\mathbf{a}}\|^2 = M$ in (3.25) leads to

$$\begin{aligned} \min_{\mathbf{a}} \mathbf{a}^*\mathbf{R}^{-1}\mathbf{a} \quad \text{subject to} \quad & \operatorname{Re}\left(\bar{\mathbf{a}}^*\mathbf{a}\right) \geq M - \frac{\epsilon}{2} \\ & \|\mathbf{a}\|^2 = M \end{aligned} \tag{3.27}$$

which is a problem that somewhat resembles the NCCB type of problem, and therefore can be solved in a similar way.

As in NCCB, when the condition

$$\operatorname{Re}\left(\bar{\mathbf{a}}^*\tilde{\mathbf{a}}\right) < M - \frac{\epsilon}{2} \tag{3.28}$$

is satisfied, equation

$$\frac{\displaystyle\sum_{m=1}^{M} \frac{|z_m|^2}{\left(\frac{1}{\gamma_m} + \hat{\lambda}\right)^2}}{\left[\displaystyle\sum_{m=1}^{M} \frac{|z_m|^2}{\left(\frac{1}{\gamma_m} + \hat{\lambda}\right)}\right]^2} = \rho \tag{3.29}$$

must be solved by a Newton's Method, using the knowledge that the solution is unique and it is lower bounded by $-1/\gamma_1$ and upper bounded by

$$\hat{\lambda} \leq \frac{\frac{1}{\gamma_M} - (M\rho)^{1/2}\left(\frac{1}{\gamma_1}\right)}{(M\rho)^{1/2} - 1}. \tag{3.30}$$

In the condition stated in equation (3.28), $\tilde{\mathbf{a}} = M^{1/2}\mathbf{u}_1 e^{j\phi}$, where $\phi = \arg\,(\mathbf{u}_1^*\bar{\mathbf{a}})$, and $\mathbf{u}_1$ is the principal eigenvector in $\mathbf{U}$ corresponding to the largest eigenvalue of $\mathbf{R}$. If the condition is not satisfied, $\hat{\sigma}_0^2 = \gamma_1/M$.

The weights of the DCRCB can be obtained using

$$\mathbf{w} = \frac{\mathbf{R}^{-1}\hat{\mathbf{a}}}{\hat{\mathbf{a}}^*\mathbf{R}^{-1}\hat{\mathbf{a}}} \tag{3.31}$$

which is equivalent to

$$\begin{aligned}\mathbf{w} = {} & \frac{1}{M - \frac{\epsilon}{2}}\left[\bar{\mathbf{a}}^*\left(\mathbf{R} + \frac{1}{\hat{\lambda}}\mathbf{I}\right)^{-1}\mathbf{R}\bar{\mathbf{a}}\right] \\ & \cdot \frac{\left(\mathbf{R} + \frac{1}{\hat{\lambda}}\mathbf{I}\right)^{-1}\bar{\mathbf{a}}}{\bar{\mathbf{a}}^*\left(\mathbf{R} + \frac{1}{\hat{\lambda}}\mathbf{I}\right)^{-1}\mathbf{R}\left(\mathbf{R} + \frac{1}{\hat{\lambda}}\mathbf{I}\right)^{-1}\bar{\mathbf{a}}}.\end{aligned} \tag{3.32}$$

The ISP can be computed as

$$\hat{\sigma}_0^2 = \frac{1}{\left(M - \frac{\epsilon}{2}\right)^2}\frac{\left[\bar{\mathbf{a}}^*\mathbf{U}(\mathbf{I} + \hat{\lambda}\mathbf{\Gamma})^{-1}\mathbf{\Gamma}\mathbf{U}^*\bar{\mathbf{a}}\right]^2}{\bar{\mathbf{a}}^*\mathbf{U}(\mathbf{I} + \hat{\lambda}\mathbf{\Gamma})^{-2}\mathbf{\Gamma}\mathbf{U}^*\bar{\mathbf{a}}}. \tag{3.33}$$

Chapter 4

Results and Discussion

In this chapter, a short introduction will be given in order to explain the problem formulation. Several approaches of source localization[1] will be studied and compared. The methods explained have been programmed using MATLAB[2]. Numerical examples are given in order to explain and clarify the differences between the approaches and also the different results that can be obtained by means of the methods. The examples are focused in estimation of the DOA and ISP, but can be extended to signal waveform estimation[3].

4.1 Problem Formulation

Consider an array comprising M sensors, and let $\mathbf{R}$ denote the theoretical covariance matrix of the array output vector. It is assumed that $\mathbf{R} > 0$ (positive definite) has the following form:

$$\mathbf{R} = \sigma_0^2 \mathbf{a}_0 \mathbf{a}_0^* + \sum_{k=1}^{K} \sigma_k^2 \mathbf{a}_k \mathbf{a}_k^* + \mathbf{Q} \tag{4.1}$$

where $(\sigma_0^2, \{\sigma_k^2\}_{k=1}^K)$ are the powers of the $(K+1)$ uncorrelated signals impinging on the array, $(\mathbf{a}_0, \{\mathbf{a}_k\}_{k=1}^K)$ are the so-called steering vectors that depend on the array geometry and are functions of the location parameters of the sources emitting the signals, $(\ )^*$ denotes the conjugate transpose, and $\mathbf{Q}$ is the noise covariance matrix. This equation is equivalent to equation (2.16),

[1] See Chapter 3 Approaches.

[2] Numerical computing environment and programming language created by The MathWorks.

[3] For further information see Chapter 5, Section 5.1 Applications.

introduced in Chapter 2, Section 2.3 Snapshot Model. It is assumed as well that the first term in (4.1) corresponds to the Signal Of Interest (SOI) and the remaining rank-one terms to K interferences. However, in the following examples the attention is centered not only in detecting one signal, but all the signals impinging in the array; although it is also discussed the behaviour of the methods, when the application demand is detecting the signal with highest incident power.

The array has half-wavelength sensor spacing. As it will be commented in Chapter 5, Section 5.2 Further Investigation, the election of this spacing determines the work frequency in a real implementation.

The *true steering vector error* is defined by $\epsilon_0 = \min_\alpha \|\mathbf{a}_0 e^{j\alpha} - \bar{\mathbf{a}}\|^2$, where $\mathbf{a}_0$ is the *true steering vector*, and $\bar{\mathbf{a}}$ is the *assumed steering vector*.

To simulate the array calibration error (the sensor amplitude and phase error as well as the sensor position error), each element of the steering vector for each incident signal is perturbed with a zero-mean circularly symmetric complex Gaussian random variable normalized so that $\epsilon_0 = 1$. The perturbing Gaussian random variables are independent of each other.

4.2 Examples

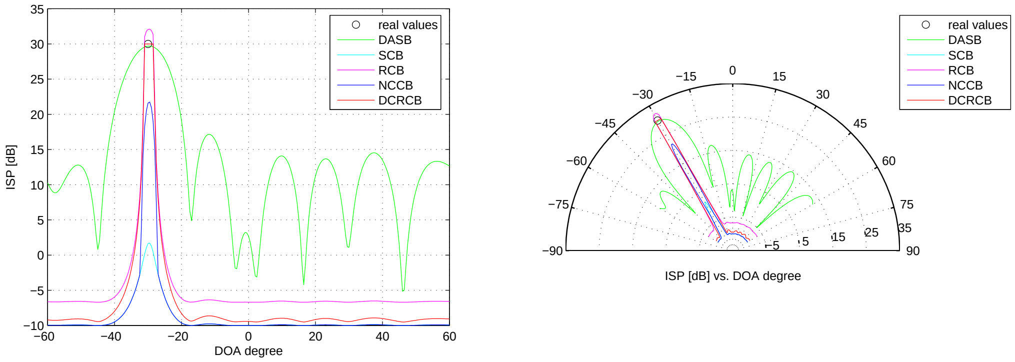
As commented before, some examples are provided in order to compare the performance of the studied beamformers in different situations, and so can judge which one is the best one depending on the application[4].

4.2.1 One Single Inciding Signal

The first example provided to compare the features of the beamformer approaches consists in a single signal inciding in an array of 10 sensors, with an angle of incidence of $-30°$ and a power of 30dB. There's a Gaussian error affecting the measures with a variance of 0.06.

As it can be seen in the figures in 4.1, all the methods point quite precisely to the DOA of the incoming signal, however they differ in the estimation of the ISP. DAS and DCRC beamformer give a more accurate estimation of the incident power, RCB gives a little too much, and NCCB and SCB are

[4] In the examples of this chapter, the Diagonal Loading method is referred as, simply, Robust Capon Beamforming.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.1: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of $-30°$ and the Gaussian error affecting the measures in the array has 0.06 variance.

the less right ones.

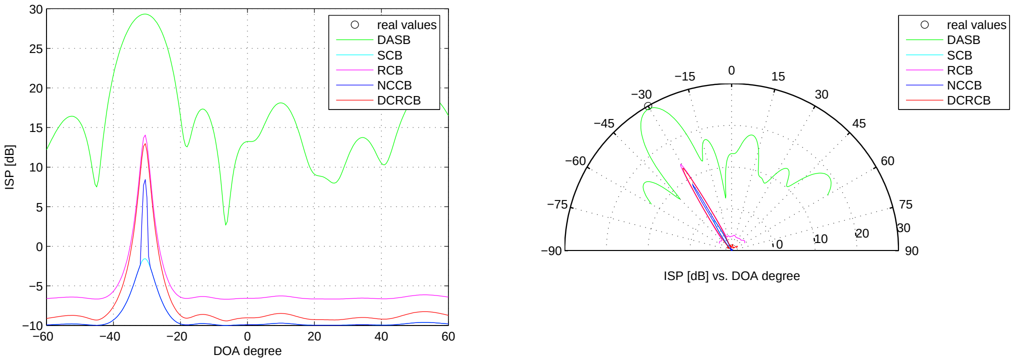
DAS beamforming presents secondary lobes that can be confused with incoming signals, this is the clear disadvantage of this method, however, if the variance is incremented to 0.1 as in the next example, figures in 4.2, DAS turns into be the most accurate in detecting the ISP.

If the noise is quite big and/or the incident power too small, as it will be seen later on, neither the DOA can be estimated with these approaches.

Next example consists in a single signal inciding in an array of 10 sensors, with an angle of incidence of $-30°$ and a power of 30dB. There's a Gaussian error affecting the measures with a variance of 0.8.

As it can be seen in the figures in 4.3, with so high noise is difficult to detect the desired incoming signal. DAS beamformer is in this case the only one which approaches more precisely to the correct ISP. This is because DAS beamformer seeks only to enhance the signal in the direction to which the array is currently steered, so if only one signal is impinging the array, it can estimate the DOA quite precisely.

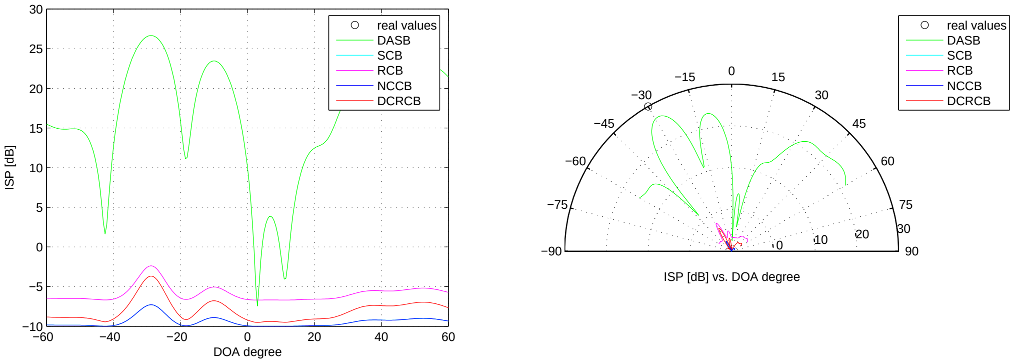
In the DAS beamformer response, it can be distinguished a bigger



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.2: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of −30° and the Gaussian error affecting the measures in the array has 0.1 variance.



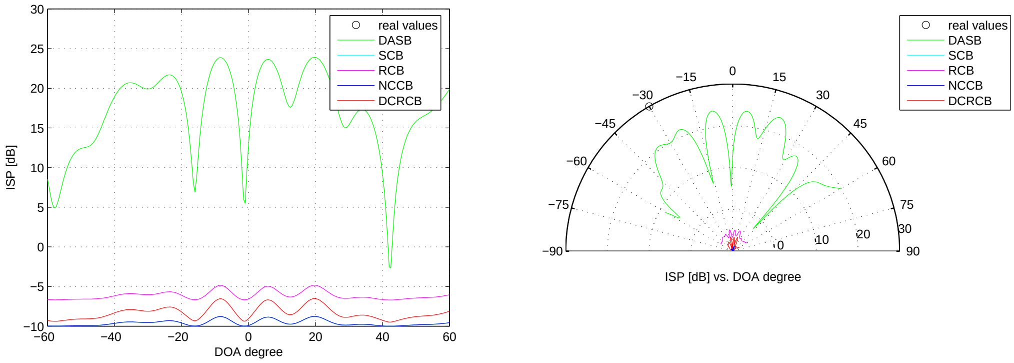
(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.3: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of −30° and the Gaussian error affecting the measures in the array has 0.8 variance. First example.

lobe that points with quite precission to the DOA. But the difference between this main lobe and the secondary one is not so big. There are several other lobes that can be confused with other incoming signals. This is a clear inconvenience of this method; when there are not only one but several signals inciding with different powers and locations, is not possible to determine all these by the number of lobes, because some of these lobes represent not inciding signals but noise.

However, with this high noise there can be the case that is not possible even to determine the DOA of the single incoming signal, as it can be seen in the figures in 4.4.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.4: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of −30° and the Gaussian error affecting the measures in the array has 0.8 variance. Second example.

In this case, neither of the studied approaches offers a solution to the problem. It is necessary to search or other methods of detecting signals with extremely low signal-to-noise ratio.

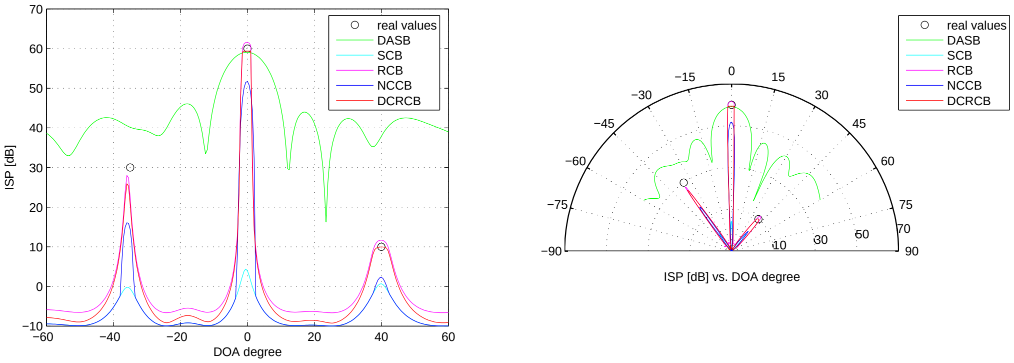
4.2.2 Several Inciding Signals

After have seen the examples with one signal, these next sections pretend to be more realistic, with not only but several signals that can be associated, in a practical implementation, with having a Signal Of Interest and some others that act as interferences.

Equally Distributed Signals

In the first example of this section, see 4.5, there is not one but several inciding signals arriving to the array of 10 sensors. These are *equally* *distributed*, what it means that the difference between the angles of arrival is approximately constant, or in other words, there are no signals ”closer” to ones rather than anothers.

The signals are arriving with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.06.



(a) Linear 2D plot.

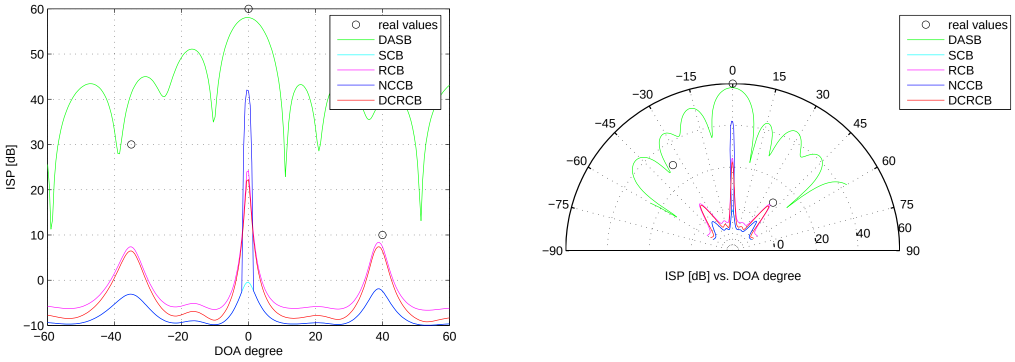
(b) Polar coordinates plot.

Figure 4.5: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.06.

In this scenario, DAS beamformer is no longer useful because it cannot detect several but only one signal, in this case, the one with higher power, as it can be seen in the represented beamformer response. However, RCB and DCRCB detect quite precissely both DOA and ISP, although RCB tends to outperform the the real power of the signals. SCB and NCCB can detect DOA but they fail to detect the incident power.

When there is more noise affecting the incoming signals, the beamformers performed as it can be seen in the figures in 4.6.

In this example, the variance of the error is of 0.1. DAS again doesn’t offer good response, and the rest of methods are not as precise as before. There is a good estimation of the DOA based on the peak locations but the estimates of the incident signal powers are way off. In this case, the highest signal power is better estimated with NCCB.

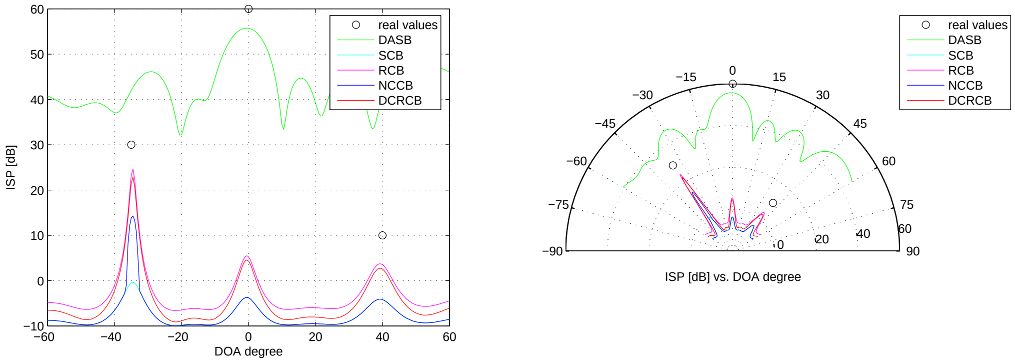


(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.6: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.1. First example.

With this high variance, it can be the case that even the methods suppress the signal with higher power, as it can be seen in the figures 4.7.

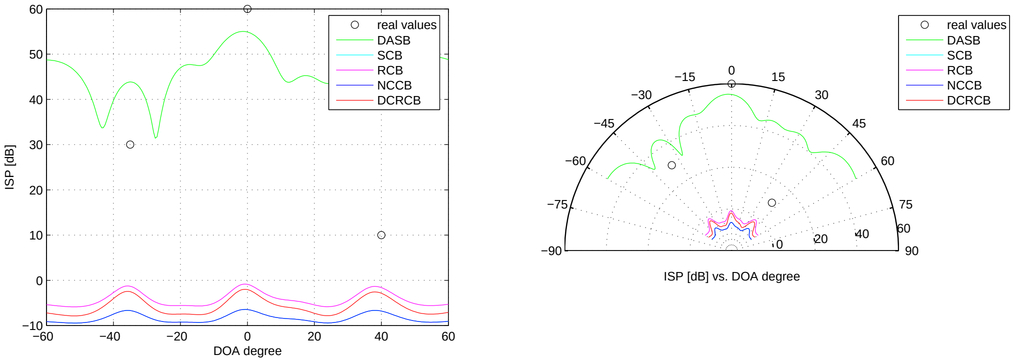


(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.7: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.1. Second example.

When the noise is extremely high, as in 4.8, and there are several signals inciding, is possible to determine the DOA but not the ISP. RCB in this case points better to the angle of incidence of the signals.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.8: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.8. First example.

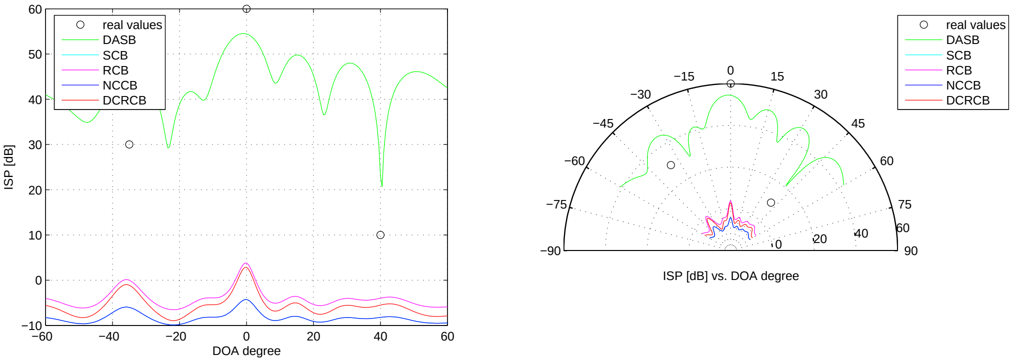
It can be possible that even DOA response tries to suppress one of the incoming signals, as it can seen in 4.9.

Close Signals

When the inciding signals are close, not only in space (there is little difference between their angles of arrival) but also when the difference between their ISPs is not big, the accuracy of the approaches decreases. The array of next examples has also 10 sensors.

In next example there are 4 inciding signals with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.06.

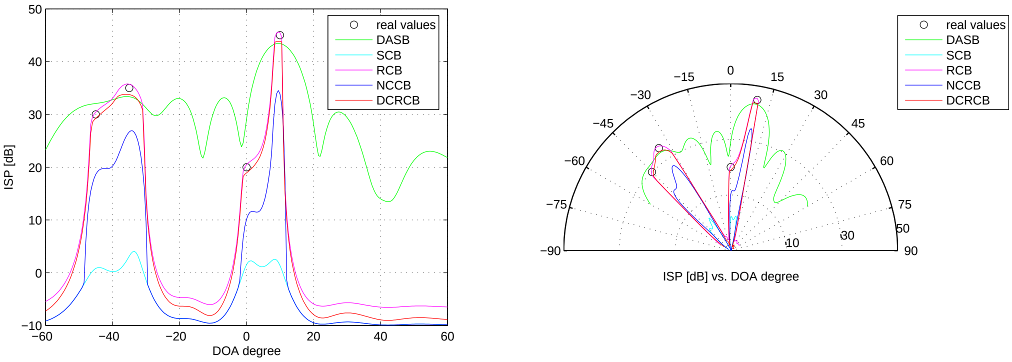
As it can be seen in 4.10, the signals with little power difference between them are hard to distinguish by the peak level with RCB or DCRCB, and it seems there is only on signal arriving with DOA around −40°. On the other hand, NCCB and SCB are capable of differenciate between the close signals; it can be appreciate peak levels around −45° and −35°, 0° and 10°,



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.9: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.8. Second example.



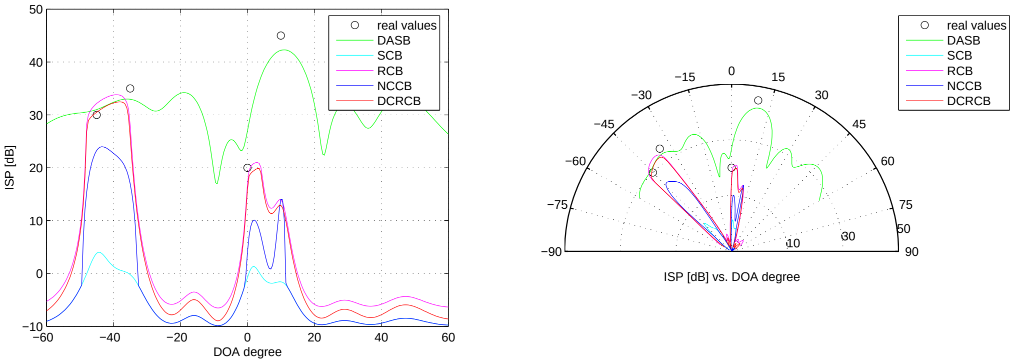
(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.10: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.06.

but the power level is way off the real one.

When the error is higher, the beamformers have the response in the figures 4.11. In this example, NCCB locate better the signals with high power difference, than the other methods, but no method is able to differenciate between the signals with little power difference.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.11: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.1. First example.

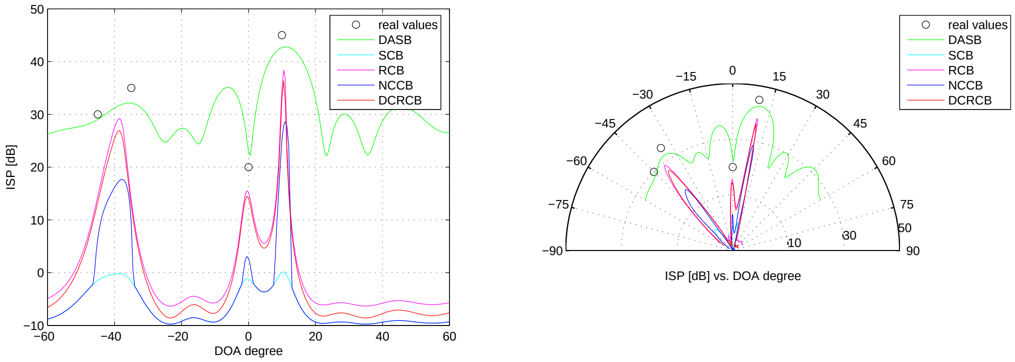
On the other hand, with this noise variance, it can be obtained a totally different result as it can be seen in 4.12. RCB and DCRCB give a good solution to the signals arriving in 0° and 10°, but it is not possible to differenciate between the other two signals.

When the noise is even higher, variance of 0.8, the response quality of the methods is really poor, as it can be seen in 4.13.

In this case, it is necessary to find some solution to improve the resolution of the methods, in order to distinguish close signals. In next section, it is proposed an *average measurements* solution, that achieves a better beamformer response when there is high noise.

If the signals are arriving with closer incident angles, the approaches cannot distinguish them as it can be seen in the figures in 4.14.

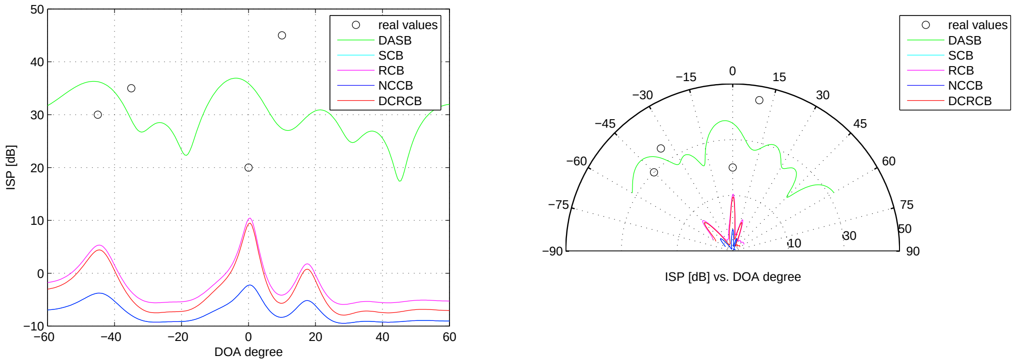
In next section it is also proposed another solution to alleviate this other problem.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.12: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.1. Second example.



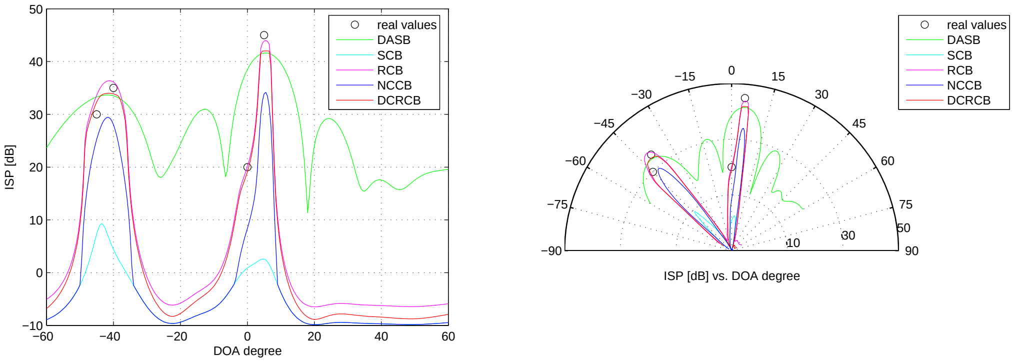
(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.13: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.8.

4.3 Improving Solutions

This section presents some ideas that can be used in a practical situation to obtain better results with these methods and be more accurate in the detection.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.14: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −40°, 0° and 5°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.06.

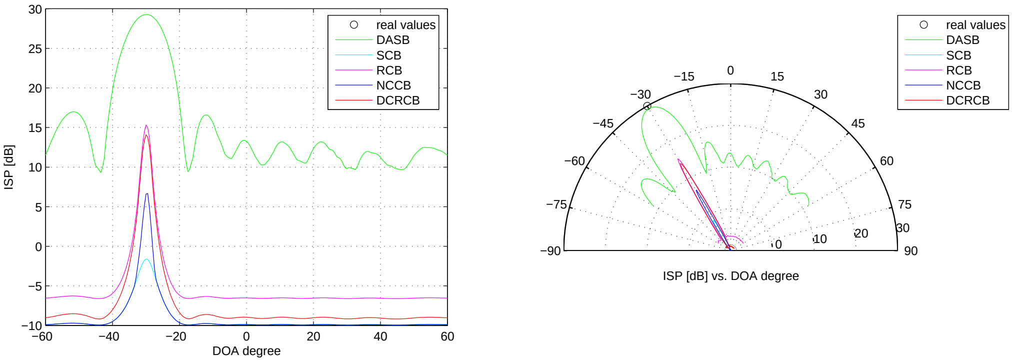
4.3.1 Average Measurements

When it comes to being accurate, one way to achieve that is taking several measurements and calculating the average of these measurements. As the nature of the noise is random, some measurements will be more accurate than others, and they will tend to compensate somehow the error.

This section includes the average measurements of the examples given in the previous section. The average has been calculated evaluating 50 times each example; each time the random noise has been different but with the same variance, trying to simulate a real scenario where the environmental conditions may change.

In the examples with only one signal arriving, when the noise was high enough the ISP was no longer detected accurately. The result of evaluating those examples with the average solution can be seen in figures 4.15 and 4.16.

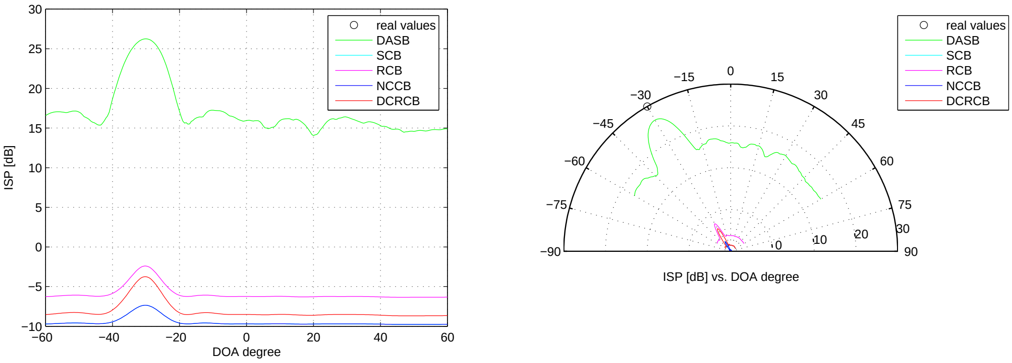
When the variance is of 0.8, taking the average really improves the response of the methods. Comparing figures in 4.3 and 4.16, it can be seen the improvement in the accuracy, especially in the DAS beamformer, whose highest peak points to the incident signal and has the best approximation to its real ISP.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.15: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of −30° and the Gaussian error affecting the measures in the array has 0.1 variance. The plots have been calculated evaluating 50 times the example and taking the average of the responses.

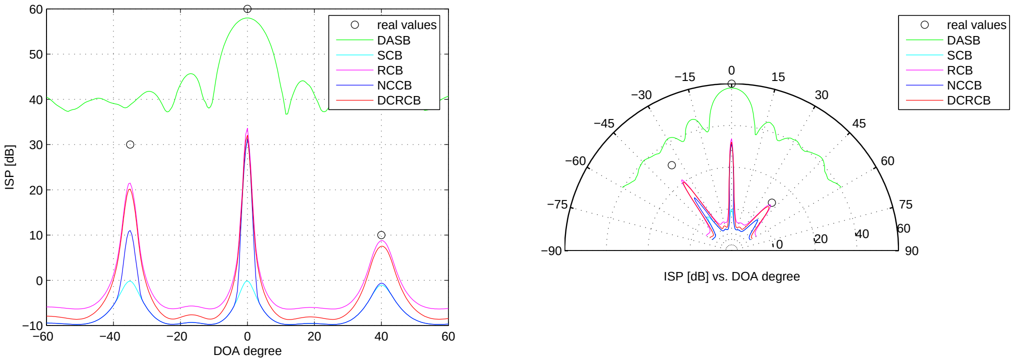


(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.16: The figures represent the response of the different beamformers to an incident signal of 30dB power. Its direction of arrival to the array of sensors is of −30° and the Gaussian error affecting the measures in the array has 0.8 variance. The plots have been calculated evaluating 50 times the example and taking the average of the responses.

In the case of several incident equally distributed signals, taking the average, as it can be seen in 4.17 and 4.18, allows a better detection of the signals. In the example with variance 0.1, RCB and DCRCB have the best performance in the peak detection; however, if the application demand is to detect just the higher power signal, the peak detection that DAS offers is more accurate in incident power than the other methods. When the noise is extremely high, like in the example with variance 0.8, the signals are slightly detected, but the estimation of the power is not correct at all. For this case it is better to find an alternative method or solution that better estimates the ISP.



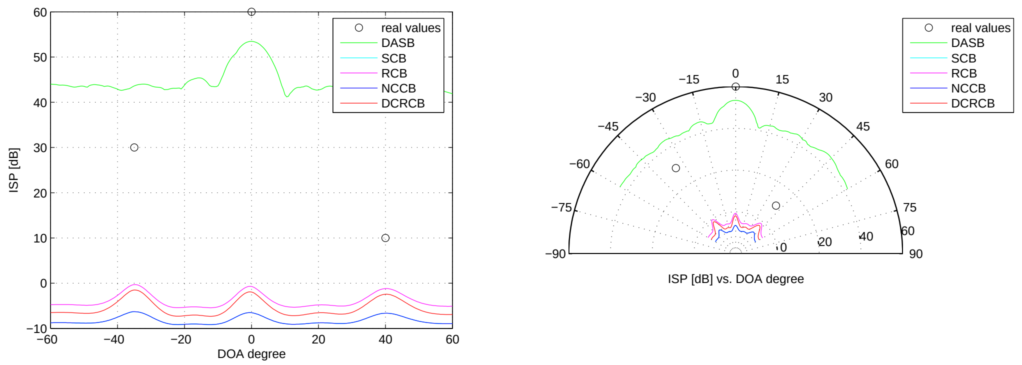
(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.17: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.1. The plots have been calculated evaluating 50 times the example and taking the average of the responses.

When the signals are not equally distributed but can be closer to ones rather than others, or in small groups, taking the average can be an advantage from just using the regular methods. It can be seen in 4.19 that the performance is better than in 4.11 or 4.12, in fact, it achieves a tradeoff between detecting close signals with little power difference and detecting close signals with high power difference.

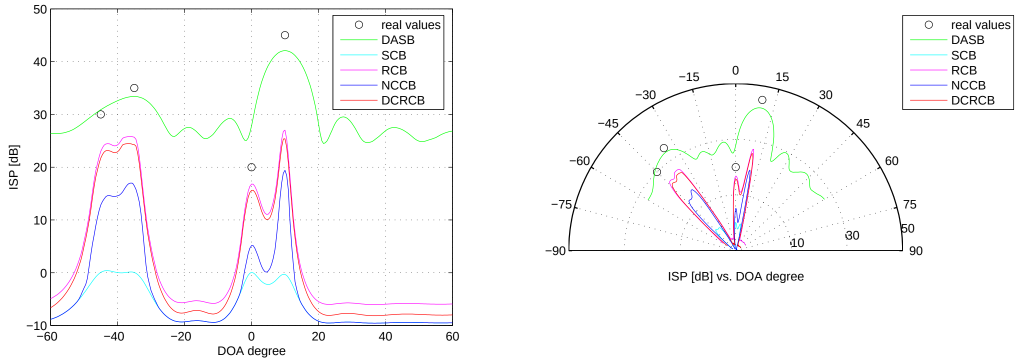
However, when the noise affecting the signals has higher variance, 0.8, taking the average does not present any advantage; this can be seen in the figures in 4.20. For this case, as commented before, it should be used an alternative method to better detect the signals. Next subsection gives an idea to improve the response of the beamformers in this sense.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.18: The figures represent the response of the different beamformers to 3 signals arriving to the array with DOA −35°, 0° and 40°, and ISP of 30, 60 and 10 dB, respectively. The variance of the Gaussian error is of 0.8. The plots have been calculated evaluating 50 times the example and taking the average of the responses.



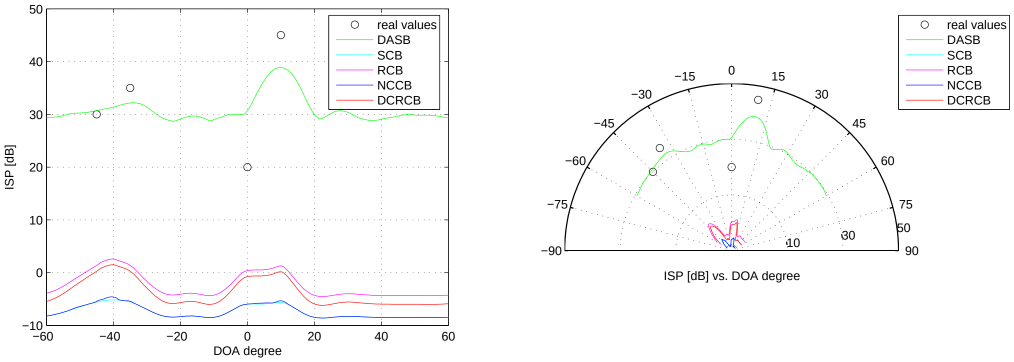
(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.19: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.1. The plots have been calculated evaluating 50 times the example and taking the average of the responses.

4.3.2 Number of Sensors

As it was seen in 4.14, when the signals are arriving with extremely close incident angles, the approaches cannot distinguish them. The proposed



(a) Linear 2D plot.

(b) Polar coordinates plot.

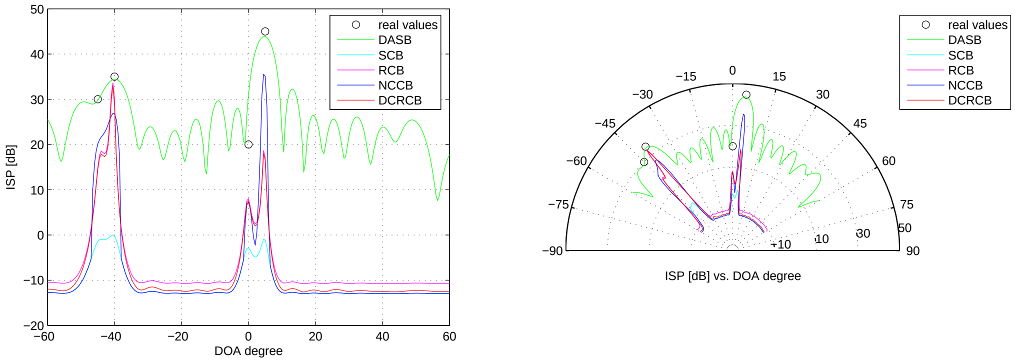
Figure 4.20: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −35°, 0° and 10°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.8. The plots have been calculated evaluating 50 times the example and taking the average of the responses.

solution to that is to increase the number of sensors in the array. By doing this, the resolution also increases, and so, closer signals can be distinguished.

Figures in 4.21 prove that with 20 receivers, the response of the beamformers is much better for the case of close signals. It can be appreciate also that NCCB gives the best approximation of the signal arriving with highest power. This can be used in applications where the main point is locating one single signal, and the rest of signals are considered interfering ones.

It was seen that when the noise variance was high, 0.8, the methods were not a good solution to discriminate the inciding signals (see 4.8, 4.9 and 4.13). Increasing the number of sensors can help in this sense as it can be seen in 4.22.

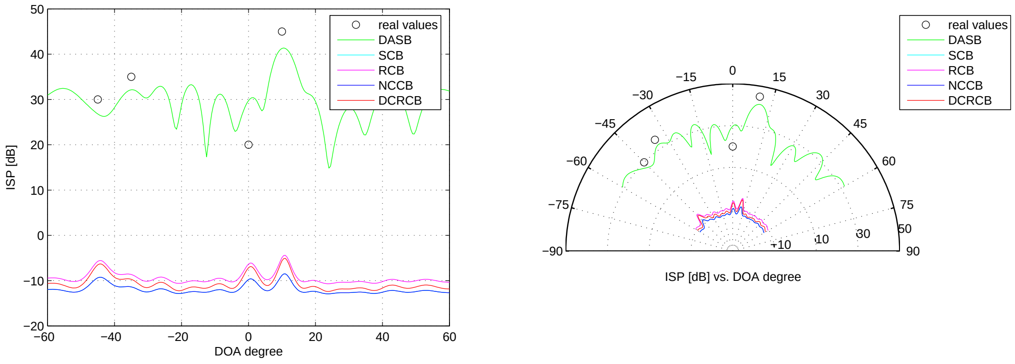
Although there is an improvement, determining the power peaks is not immediately clear. Increasing even more the number of sensors, as in the figures in 4.23 gives a better result in peak detection, but the power detection is way off. With this example it can be seen the existing **duality DOA-ISP**. This will be explained later on, in next chapter. Also there is an underlying problem with this technique, because it is not always possible to use a big number of sensors in the array. This will be discussed later on, as well.



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.21: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −40°, 0° and 5°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.06 and the number of sensors is 20.

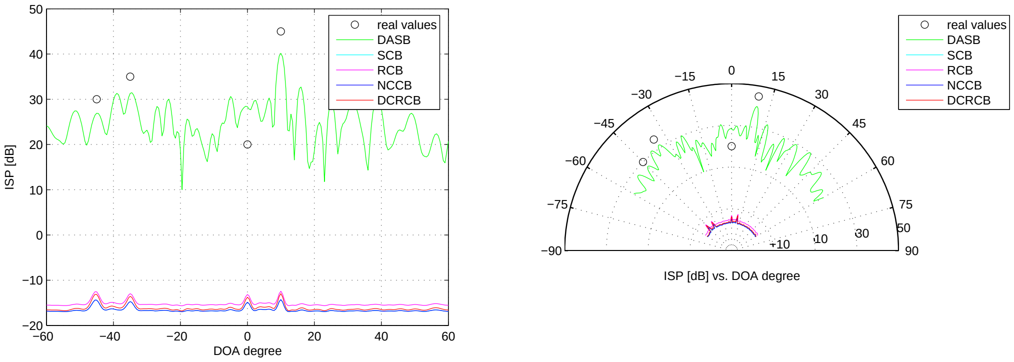


(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.22: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −40°, 0° and 5°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.8 and the number of sensors is 20.

Anyway, combining both explained techniques, taking the average, and increasing the number of sensors, can give good results in adverse environmental situations. But it is important not to forget the practical application behind all the theory, because it determines the solutions that must be



(a) Linear 2D plot.

(b) Polar coordinates plot.

Figure 4.23: The figures represent the response of the different beamformers to 4 signals arriving to the array with DOA −45°, −40°, 0° and 5°, and ISP of 30, 35, 20 and 45 dB, respectively. The variance of the Gaussian error is of 0.8 and the number of sensors is 50.

applied in each case.

Chapter 5

Findings and Future Investigation

The aim of this chapter is being a summary of the realized work; extracting the main concluding ideas in an organized way for the reader. This chapter also points out the lines that can be follow in a future to continue the development of the work.

5.1 Applications

As it has been said several times through the work, the final application which is the goal of the beamforming, determines which method is more suitable for the proposit. Here, the work has been centered in DOA and ISP estimation, which are the important parameters when reciving a signal, that must be taken into account to locate the source position. The DOA gives the direction from which the source is emitting, and the ISP determines how far it is, i. e., the range. In general lines, more ISP indicates a closer source, and the other way round; but to accurately determine the range, it is necessary to use matched-field processing, in which the existing models for the waves propagation can be used directly. But this is beyond the scope of this work and can be considered as a future line of investigation.

The final application determines the method to use; that has been shown by the examples. When there was only one signal impinging in the array, and the goal was to determine the DOA and ISP of it, RCB was quite precise in DOA estimation, although DAS gave the most accurate received power (by looking to the power peak level).

However, this situation is an idealized one and this conclusions must not be used in a practical implementation. In a real scenario there can be always interferences; they cannot be controlled, so it is not possible to trust in having only one inciding signal. Because of this, the *robust methods* are used.

There are several observations that have been concluded by the examples. When the difference between the powers of the inciding signals was big and there was quite noise, the methods were not working properly. That is, it was not possible to determine the ISP of the high power signal and of the low power one, with enough accuracy. This can be also another line of further investigation. However, if the application goal is not to detect all the signals but there is only one SOI, the methods are giving a good approximation in most of the cases.

In general through the work, the DOA has been estimated quite precisely, but there have been always problems when estimating the ISP. In other words, it has been easier to say in which direction was the source than how far it was. In fact, there is a **duality ISP accuracy — DOA accuracy** which means that there must be a tradeoff between the detection of ISP and DOA; trying to obtain both as precisely as possible, but knowing that if the effort is made in detecting one, the other one will be less accurate. In the work, there were some examples where the number of sensors in the array was high; this was a solution to fight the noise, but the power was way off. Also, using a high number of sensors entail some other problems, as it will be commented later on.

5.2 Conclusions

This section summarizes the conclusions extracted through this work. It is centered in explaining where and when each method should be used depending on the demands of the application.

DCRCB is the preferred choice for applications requiring high signal-to-noise ratio, whereas RCB is the favored one for applications demanding accurate signal power estimation. As this paper wants to be focused in estimating range and DOA; RCB is the best election.

SCB is clearly more affected by the noise than the rest of methods; when in the examples the noise was being increased, SCB was the first one that was not detecting the incident power properly.

As said before, big differences in the incident power of the signals make that the higher power signal peaks are not well estimated. However when the signals are about the same incident power but are inciding with a similar direction of arrival, it is not possible to determine if there is one signal arriving or several ones, and the response of the beamformers is just one wide peak. This problem can be solved increasing the number of sensors in the array to increase also the resolution.

NCCB is the method that better works to detect one single signal, when this signal has strong power and the rest interfering ones are arriving with little power. This approach is the one that better detects the power, i. e., the range of the inciding signal. Because of this, it can be used when the application demand is detecting just one signal.

With high noise and several inciding signals, the beamformers responses show that sometimes the signals with little difference power are well detected but not the ones with high difference power, and sometimes it happens just the other way round. This is dependant on the noise, and can be alleviate by using several measures and taking the average.

Whith extremely high noise it is difficult to detect the incident signals. Increasing the number of sensors can help in the detection, but also presents the problem that the detection of the ISP is less accurate.

Anyway, combining both explained techniques, taking the average, and increasing the number of sensors, can give good results in adverse environmental situations. But it is important not to forget the practical application behind all the theory, because it determines the solutions that must be applied in each cases.

To help the reader to better understand the conclusions, it has been chosen to summarize the main findings in tables. The first table, see 5.1, shows the better approach in each case when the noise is low, taking into account that the goals are DOA and ISP detection, so the method must be a tradeoff between them. To arrive to this conclusions it has been also considered the use of the improving solutions to have more accurate responses.

Table 5.2 shows also the better approach to use in each case, but when the noise is high.

Table 5.1: Table showing the most suitable approach in each case. Low noise.

Single Signal	Several Signals		
	Equally-Spaced	Close	
		Big Diff. Power	Little Diff. Power
DCRCB	**DCRCB**	**NCCB**	**DCRCB**

Table 5.2: Table showing the most suitable approach in each case. High noise.

Single Signal	Several Signals		
	Equally-Spaced	Close	
		Big Diff. Power	Little Diff. Power
DAS	**RCB**	**RCB**	**RCB**

5.3 Further Research

As it has been said, in general lines more ISP indicates a closer source, and the other way round; but to accurately determine the range, it is necessary to use matched-field processing, in which the existing models for the waves propagation can be used directly. This can be a future research line; matching the obtained results with a propagation model, so they can be used in a practical application.

The future research can also be centered in finding other applications in which the beamformers can be used. In many applications, such as in communications or the global positioning system, the focus is on SOI waveform estimation. This is different from what it has been studied, but the work can also be used and extended to this field.

In the work, there were some examples where the number of sensors in the array was high. This was a solution to fight the high noise and detecting in an accurate way the DOA, but there is an evident problem; in a physical implementation it is not possible to have as many sensors as the theoretical simulations demands, because of problems of space, and also of monetary cost. So once more, it is necessary to find a tradeoff between number of sensors (accuracy) and cost (space, money).

The election of the spacing in the array determines the work frequency in a practical implementation. This limitates the applications; the sensor will be calibrated to a certain frequency, and it will be costly changing to another one. There can be an investigation in this direction, to study the relation between the array spacing and the detecting frequency, the effects of changing it, the possibility of reuse the same array with several frequencies by just changing some parameters in the algorithms, etc.

Further investigation can be related to the array geometry. In a practical implementation having a 2D-array supposes an advantage in precission. With the studied 1D-array the angle of incidence was obtained, but of course, this angle was contained in the perpendicular plane to the line of sensors. A surface of arrays gives the possibility of detecting sources in the space.

After taking into account all this theoretical results, the next logical step is to implement the application. It is necessary first to look for an appropiate type of sensor to use, matching its characteristics with the requirements the approaches need.

References

[1] B. D. Van Veen and K. M. Buckley, “Beamforming: A Versatile Approach to Spatial Filtering,” *IEEE ASSP Magazine*, vol. 5, pp. 4–24, april 1988.

[2] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part IV, Optimum Array Processing*, vol. 4. John Wiley & Sons, april 2002.

[3] A. B. Gershman, “Robust Adaptive Beamforming: an Overview of Recent Trends and Advances in the Field,” in *International Conference on Antenna Theory and Techniques*, (Sevastopol, Ukraine), pp. 30–35, IEEE, september 2003.

[4] A. B. Gershman, T. Kirubarajan, and A. El-Keyi, “Robust Adaptive Beamforming Based on the Kalman Filter,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 53, august 2005.

[5] J. Lin, Q. Peng, and H. Shao, “On Diagonal Loading for Robust Adaptive Beamforming Based on Worst-Case Performance Optimization,” *ETRI Journal*, vol. 29, february 2007.

[6] J. Li, P. Stoica, and Z. Wang, “On Robust Capon Beamforming and Diagonal Loading,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 51, july 2003.

[7] J. Li, P. Stoica, and Z. Wang, “Doubly Constrained Robust Capon Beamformer,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 52, september 2004.

[8] H. Krim and M. Viberg, “Two Decades of Array Signal Processing Research. The Parametric Approach,” *IEEE SIGNAL PROCESSING MAGAZINE*, july 1996.

[9] P. Stoica, Z. Wang, and J. Li, “Robust Capon Beamforming,” *IEEE SIGNAL PROCESSING LETTERS*, vol. 10, june 2003.

[10] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* New York: Wiley, 1968.

[11] D. Hanselman, “Fully Customizable Polar Plot.” MATLAB Central File Exchange, january 2005. `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6681&objectType=File`.

Appendix A

MATLAB Functions

This appendix includes the complete code of all the MATLAB functions used through this work.

A.1 Approaches

The different approaches code.

A.1.1 Delay And Sum Beamformer

```
% Syntax:
% [theta,ISPdB] = DASB_ISPdB(M,aR);
%
% This MATLAB function simulates a Delay And Sum Beamformer.
% It obtains the Incident Signal Power as a function of the
% angle of arrival, theta.
%
% Input:
% M     - number of receivers in the array
% aR    - sample covariance matrix
%
% Output:
% theta - angle of arrival, theta (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
```

```
%
% M = 10;


function [theta,ISPdB] = DASB_ISPdB(M,aR);

beta = 1;
% When beta = 1, NCCB becomes the DAS beamformer, and hence, it
% uses the assumed array steering vector divided by M as the
% weight vector.

i = 1;
ISP = zeros(1,241);

for theta = -60+90:1/2:60+90,

    theta = theta*pi/180;

    a = ones(M,1);
    for rec = 1:M,
        a(rec) = exp(-j*(rec-1)*pi*cos(theta));
    end

    w = a./M;
    wct = w';

    % Incident Signal Power:
    ISP(i) = wct*aR*w;

    i = i + 1;

end

theta = -60:1/2:60;
ISPdB = 10*log10( real(ISP) );

% --------------------------------------------------------------
```

A.1.2 Standard Capon Beamformer

```
% Syntax:
% [theta,ISPdB] = SCB_ISPdB(M,aR);
%
% This MATLAB function simulates a Standard Capon Beamformer.
% It obtains the Incident Signal Power as a function of the
% angle of arrival, theta.
%
% Input:
% M     - number of receivers in the array
% aR    - sample covariance matrix
%
% Output:
% theta - angle of arrival, theta (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
%
% M = 10;


function [theta,ISPdB] =  SCB_ISPdB(M,aR);

% Compute the eigendecomposition of R (or, in practice, of ^R):
[U,Gamma,V] = svd(aR);
Uct = U';

% We know that R^(-1) = U*(1/Gamma)*U'.
R_1 = U*diag(1./diag(Gamma))*Uct;

i = 1;
ISP = zeros(1,241);

for theta = -60+90:1/2:60+90,

    theta = theta*pi/180;

    a = ones(M,1);
    for rec = 1:M,
        a(rec) = exp(-j*(rec-1)*pi*cos(theta));
    end

    act = a';

    % Incident Signal Power:
    ISP(i) = 1/(act*R_1*a);

    i = i + 1;

end

theta = -60:1/2:60;
ISPdB = 10*log10( real(ISP) );

% ---------------------------------------------------------------
```

A.1.3 Diagonal Loading

```
% Syntax:
% [theta,ISPdB] = RCB_ISPdB(M,aR);
%
% This MATLAB function simulates a Robust Capon Beamformer.
% It obtains the Incident Signal Power as a function of the
% angle of arrival, theta.
% It uses the function 'lambda310.m'.
%
% Input:
% M     - number of receivers in the array
% aR    - sample covariance matrix
%
% Output:
% theta - angle of arrival, theta (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
%
```

```
% M = 10;


function [theta,ISPdB] = RCB_ISPdB(M,aR);

theta = 0;
ISPdB = 0;

% Step 1) Compute the eigendecomposition of R (or more
% practically of ^R).

[U,Gamma,V] = svd(aR);
Uct = U';

gamma = zeros(1,M);
gamma(1) = Gamma(1,1);
for i = 2:M,
    gamma(i) = Gamma(i,i);
end
gamma = gamma.';
% Example with M = 10:
%gamma = [Gamma(1,1) Gamma(2,2) Gamma(3,3) Gamma(4,4)...
%Gamma(5,5) Gamma(6,6) Gamma(7,7) Gamma(8,8) Gamma(9,9)...
%Gamma(10,10)].';

% Step 2) Solve equation (3.10) for landa, e.g., by a Newton's
% method, using the knowledge that the solution is unique and
% it belongs to the interval in (3.11).

epsilon = 1.0;

i = 1;
ISP = zeros(1,241);

for theta = -60+90:1/2:60+90,

    theta = theta*pi/180;

    a = ones(M,1);
    for rec = 1:M,
        a(rec) = exp(-j*(rec-1)*pi*cos(theta));
    end

    act = a';

    z = Uct*a;

    % Lower bound on lambda:
    lblambda = ( norm(a) - sqrt(epsilon) )/...
        ( gamma(1)*sqrt(epsilon) );
    % Upper bound on lambda:
    ublambda = min([ ( ((1/epsilon)*sum((abs(z).^2)./...
        gamma.^2))^(1/2) )( (norm(a)-sqrt(epsilon))/...
        (gamma(M)*sqrt(epsilon)) ) ]);

    [lambda] = lambda310(epsilon,z,gamma,lblambda,ublambda);

    % Step 3)

    dmatrix = eye(M) + lambda*Gamma;
    idmatrix = diag(1./diag(dmatrix));

    a0 = a - U*idmatrix*Uct*a;

    % Step 4)

    longdmatrix = lambda^(-2)*eye(M) + 2*lambda^(-1)*Gamma +...
        Gamma^2;
    ilongdmatrix = diag(1./diag(longdmatrix));

    ISP(i) = 1 / ( act*U*Gamma*ilongdmatrix*Uct*a ) ;

    i = i + 1;

end

theta = -60:1/2:60;
ISPdB = 10*log10( real(ISP) );

% ---------------------------------------------------------------
```

A.1.4 Norm Constrained Capon Beamformer

```
% Syntax:
% [theta,ISPdB] = NCCB_ISPdB(M,aR);
%
% This MATLAB function simulates a Norm Constrained Capon
% Beamformer. It obtains the Incident Signal Power as a
% function of the angle of arrival, theta.
% It uses the function 'lambda319.m'.
%
% Input:
% M     - number of receivers in the array
% aR    - sample covariance matrix
%
% Output:
% theta - angle of arrival, theta (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
%
% M = 10;

function [theta,ISPdB] = NCCB_PvsTheta(M,aR);

% Step 1) Compute the eigendecomposition of R (or, in practice,
% of ^R).

[U,Gamma,V] = svd(aR);

% Step 2) If (3.20) is satisfied, solve (3.19) for ^landa,
% e.g., by a Newton's method, using the knowledge that the
% solution is unique and it is lower bounded by 0 and upper
% bounded by (3.21); otherwise, set ^landa = 0.

% (3.20)  ==  zeta < cond

beta = 6.0;
% so that the peak widths of the NCCB and DCRCB are about the
% same.
```

```
zeta = beta / M;

Uct = U';

dGamma = diag(Gamma);
R_1 = U*diag(1./dGamma)*Uct;
R_2 = R_1*R_1;

% (3.21)

% Lower bound on lambda:
lblambda = 0;
% Upper bound on lambda:
ublambda = ( dGamma(1) - ((M*zeta)^(1/2))*dGamma(M) )/...
    ( ((M*zeta)^(1/2)) - 1 );

i = 1;
ISP = zeros(1,241);

for theta = -60+90:1/2:60+90,

    theta = theta*pi/180;

    a = ones(M,1);
    for rec = 1:M,
        a(rec) = exp(-j*(rec-1)*pi*cos(theta));
    end

    act = a';
    cond = real( (act*R_2*a)/(act*R_1*a)^2 );

    z = Uct*a;

    if zeta < cond
        [lambda]=lambda319(M,zeta,z,dGamma,lblambda,ublambda);
    else
        lambda = 0;
    end

    dmatrix = Gamma + lambda*eye(M);
    idmatrix = diag(1./diag(dmatrix));
    w = U*idmatrix*z / (act*U*idmatrix*z);
    wct = w';

    % Incident Signal Power:
    ISP(i) = wct*aR*w;

    i = i + 1;

end

theta = -60:1/2:60;
ISPdB = 10*log10( real(ISP) );

% -------------------------------------------------------------
```

A.1.5 Doubly Constrained Robust Capon Beamformer

```
% Syntax:
% [theta,ISPdB] = DCRCB_ISPdB(M,aR);
%
% This MATLAB function simulates a Doubly Constrained Robust
% Capon Beamformer.
% It obtains the Incident Signal Power as a function of the
% angle of arrival, theta.
% It uses the function 'lambda329.m'.
%
% Input:
% M     - number of receivers in the array
% aR    - sample covariance matrix
%
% Output:
% theta - angle of arrival, theta (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
%
% M = 10;
```

```
function [theta,ISPdB] = DCRCB_ISPdB(M,aR);

theta = 0;
ISPdB = 0;

% Step 1) Compute the eigendecomposition of R (or, in practice,
% of ^R).

[U,Gamma,V] = svd(aR);
Uct = U';

% Step 2) If (3.28) is satisfied, solve (3.29) for ^landa,
% e.g., by a Newton's method, using the knowledge that the
% solution is unique and it is lower bounded by -1/gamma1 and
% upper bounded by (3.30), and then continue to Step 3;
% otherwise, compute ^sigma0_2 = gamma1/M and stop.

% Let u1 denote the first eigenvector in U:
u1 = U(:,1);

epsilon = 1.0;
cond = M - epsilon/2;
rho = M/(cond^2);

% A vector containing the main diagonal of inv(Gamma):
iGamma = 1./diag(Gamma);

% Lower bound on lambda:
lblambda = -iGamma(1);
% Upper bound on lambda:
ublambda = ( iGamma(M) - ((M*rho)^(1/2))*iGamma(1) )/...
    ( ((M*rho)^(1/2)) - 1 );


i = 1;
s_a = zeros(1,241);
ISP = zeros(1,241);

for theta = -60+90:1/2:60+90,
    theta = theta*pi/180;

    a = ones(M,1);
    for rec = 1:M,
        a(rec) = exp(-j*(rec-1)*pi*cos(theta));
    end

    act = a';

    % angle(x) returns the phase angles, in radians, of a
    % matrix with complex elements:
    phi = angle(u1'*a);

    est_a = M^(1/2)*u1*exp(j*phi);

    R_a_times_a = real(act*est_a);

    z = Uct*a;

    % (3.28) == Re(a*·est_a) < M - epsilon/2

    if R_a_times_a < cond
        [lambda] = lambda329(M,rho,z,iGamma,lblambda,ublambda);
    else
        lambda = 0;
        ISP(i) = Gamma(1,1)/M;
        % Continue to the next iteration:
        i = i + 1;
        continue
    end

    dmatrix = eye(M) + lambda*Gamma;
    idmatrix = diag(1./diag(dmatrix));
    idmatrix2 = idmatrix*idmatrix;

    ISP(i) = (1 / cond^2) * ((act*U*idmatrix*Gamma*z)^2) /...
        ( act*U*idmatrix2*Gamma*z );
    i = i + 1;

end
```

```
theta = -60:1/2:60;
ISPdB = 10*log10( real(ISP) );

% ---------------------------------------------------------------
```

A.2 Auxiliar Functions

This section includes the auxiliar functions, as the ones used to plot and compare the results obtained with the different approaches[1], and the ones that implement a Newton's Method.

A.2.1 Plotting

```
% Syntax:
% [DOA,ISPdB] = ISPdBvsDOA(M,var,theta,sigma);
%
% This MATLAB function obtains and plots the Incident Signal
% Power (ISP) as a function of the Direction Of Arrival (DOA).
% It uses the functions 'ArrayRec.m' and 'mmpolar.m'.
%
% Input:
% M     - number of receivers in the array
% var   - variance of the white Gaussian error produced in the
%         array
% theta - row vector containing the DOA of the different
%         sources (degrees -90 to +90)
% sigma - row vector containing the ISP of the different
%         sources (dB)
%
% Output:
% DOA   - Direction Of Arrival (degrees -90 to +90)
% ISPdB - Incident Signal Power (dB)
%
%
% EXAMPLE1:
%
% M = 10;
```

[1] For information about the function ‘mmpolar.m’ see reference [11].

```
% var = 0.06;
% theta = [-35 -15 0 10 40];
% sigma = [30 60 40 35 10];


function [DOA,ISPdB] = ISPdBvsDOA(M,var,theta,sigma);

DOA = 0;
ISPdB = 0;

aR = ArrayRec(M,var,theta,sigma);
% ---------------------------------------------------------------

[theta1,ISPdB1] = DASB_ISPdB(M,aR);
[theta2,ISPdB2] = SCB_ISPdB(M,aR);
[theta3,ISPdB3] = RCB_ISPdB(M,aR);
[theta4,ISPdB4] = NCCB_ISPdB(M,aR);
[theta5,ISPdB5] = DCRCB_ISPdB(M,aR);

DOA = [theta1 ; theta2 ; theta3; theta4; theta5];
ISPdB = [ISPdB1 ; ISPdB2 ; ISPdB3; ISPdB4; ISPdB5];

% Plotting:

close all hidden

clf;
hold on;
box on;
xlabel('DOA (angle \theta in degrees)');
ylabel('ISP [dB]');

% The real signals arriving to the array:
plot(theta,sigma,'ok');

plot(theta1,ISPdB1,'g');
plot(theta2,ISPdB2,'c');
plot(theta3,ISPdB3,'m');
plot(theta4,ISPdB4);
plot(theta5,ISPdB5,'r');
```

```
h = legend('real values','DASB','SCB','RCB','NCCB','DCRCB');

thetapolar = theta.*pi./180;
thetapolar1 = theta1.*pi./180;
thetapolar2 = theta2.*pi./180;
thetapolar3 = theta3.*pi./180;
thetapolar4 = theta4.*pi./180;
thetapolar5 = theta5.*pi./180;

figure;

% The real signals arriving to the array:
mmpolar(thetapolar,sigma,'ok');
hold on;

mmpolar(thetapolar1,ISPdB1,'g');
mmpolar(thetapolar2,ISPdB2,'c');
mmpolar(thetapolar3,ISPdB3,'m');
mmpolar(thetapolar4,ISPdB4);
mmpolar(thetapolar5,ISPdB5,'r');

mmpolar('TTickDelta',15,'TLimit',[-pi/2 pi/2]);
view(90,-90);
ylabel('ISP [dB] as a function of the DOA [degrees]');

k = legend('real values','DASB','SCB','RCB','NCCB','DCRCB');

% --------------------------------------------------------------
```

```
% Syntax:
% aR = ArrayRec(M,var,theta,sigma);
%
% This MATLAB function simulates the ARRAY OF RECEIVERS.
% It obtains the SAMPLE COVARIANCE MATRIX as a function of the
% number of receivers (M), the direction or angle of arrival
% (theta) and the received power (sigma), adding an error
% (white Gaussian noise) that simulates the finite sample
% support, imperfect array calibration, etc.
% It uses the function 'aassumed.m'.
```

```
%
% Input:
% M     - number of receivers in the array
% var   - variance of the white Gaussian error produced in the
%         array
% theta - row vector containing the DOA of the different
%         sources (degrees -90 to +90)
% sigma - row vector containing the ISP of the different
%         sources (dB)
%
% Output:
% aR    - sample covariance matrix
%
%
% EXAMPLE1:
%
% M = 10;
% var = 0.06;
% theta = [-35 -15 0 10 40];
% sigma = [30 60 40 35 10];


function aR = ArrayRec(M,var,theta,sigma);

aR = 0;

% Number of signals arriving to the receiver:
n = max(size(theta));

% Signal directions in degrees (0 to +180):
thetad = theta + 90;

% Signal directions in radians:
thetar = thetad.*pi./180;

% Signal powers (linear):
sigmal = 10.^(sigma./10);

% TRUE Steering vectors:
a = ones(n,M);
for i = 1:n,
    for rec = 1:M,
        a(i,rec) = exp(-j*(rec-1)*pi*cos(thetar(i)));
    end
end

a = a.';

% We define the true steering vector ERROR
%   epsilon0 = min_alpha norm( a*exp(j*alpha) - aa )^2 ,
% where a is the true steering vector, and aa is the assumed
% one.

% ASSUMED Steering vectors:
aa = aassumed(var,a);

% ASSUMED Steering vectors (conjugate transpose):
aact = aa';

% Noise covariance matrix:
Q = eye(M);

% Theoretical covariance matrix of the array output vector
% (assumed one):
aR = Q;
for i = 1:n,
    aR = aR + sigmal(i)*aa(:,i)*aact(i,:);
end

% ---------------------------------------------------------------
```

```
% Syntax:
% aa = aassumed(var,a);
%
% This MATLAB function obtains the real array manifold vector.
% Each element of the steering vector for each incident signal
% is perturbed with a zero-mean circularly symmetric complex
% Gaussian random normalized variable.
%
% Input:
```

```
% var   - variance of the white Gaussian error produced in the
%         array
% a     - array manifold vector (theoretical one)
%
% Output:
% aa    - array manifold vector (real one)
%
% EXAMPLE1:
%
% var = 0.06;


function aa = aassumed(var,a);

[M,n] = size(a);
% M - number of sensors.
% n - number of signals.

z = sqrt(var)*( randn(M,n) + j*randn(M,n) );

aint = a + z;

aa = aint./norm(aint).*sqrt(M);

% -------------------------------------------------------------
```

A.2.2 Newton's Method

```
% This program solves (3.10) by the Newton's Method

function lambda0 = lambda310(epsilon,z,gamma,lblambda,ublambda);

% Initial guess of lambda:
lambda0 = 0;

dif = 1;
while dif > 1e-12,

    g = sum( abs(z).^2 ./ (1 + lambda0*gamma).^2 );
    f = g - epsilon;

    f_der = -2 * sum( gamma.*(abs(z).^2) ./...
        (1+lambda0*gamma).^3 );

    lambda1 = lambda0 - f/f_der;
    dif = abs(lambda1 - lambda0);
    lambda0 = lambda1;

end

% IS IT WORKING?
%g = sum( abs(z).^2 ./ (1 + lambda0*gamma).^2 );
%f = g - epsilon

if (lambda0 < lblambda) | (lambda0 > ublambda),
    lambda0 = NaN;
end

% ---------------------------------------------------------------



% This program solves (3.19) by the Newton's Method

function [lambda0]=lambda319(M,zeta,z,Gamma,lblambda,ublambda);

% Initial guess of lambda:
lambda0 = 0;

dif = 1;
while dif > 1e-9,

    numerator = sum( abs(z).^2 ./ (Gamma + lambda0).^2 );
    denominator = (sum( abs(z).^2 ./ (Gamma + lambda0) )).^2;
    f = numerator/denominator - zeta;

    num_der = -2 * sum( abs(z).^2 ./ (Gamma + lambda0).^3 );
    den_der = 2 * sum( abs(z).^2 ./ (Gamma +lambda0) ) * ...
        ( -sum( abs(z).^2 ./ (Gamma + lambda0).^2 ) );
    f_der = ( num_der*denominator - numerator*den_der )...
```

```
        / denominator^2;

    lambda1 = lambda0 - f/f_der;
    dif = abs(lambda1 - lambda0);
    lambda0 = lambda1;

end

% IS IT WORKING?
%numerator = sum( abs(z).^2 / (Gamma + lambda0).^2 );
%denominator = (sum( abs(z).^2 / (Gamma + lambda0) )).^2;
%f = numerator/denominator - zeta

if (lambda0 < lblambda) | (lambda0 > ublambda),
    lambda0 = NaN;
end

% -------------------------------------------------------------



% This program solves (3.29) by the Newton's Method

function [lambda0]=lambda329(M,rho,z,iGamma,lblambda,ublambda);

% Initial guess of lambda:
lambda0 = 0;

dif = 1;
while dif > 1e-9,

    numerator = sum( abs(z).^2 ./ (iGamma + lambda0).^2 );
    denominator = (sum( abs(z).^2 ./ (iGamma + lambda0) )).^2;
    f = numerator/denominator - rho;

    num_der = -2 * sum( abs(z).^2 ./ (iGamma + lambda0).^3 );
    den_der = 2 * sum( abs(z).^2 ./ (iGamma +lambda0) ) *...
        ( -sum( abs(z).^2 ./ (iGamma + lambda0).^2 ) );
    f_der = ( num_der*denominator - numerator*den_der ) /...
        denominator^2;
    lambda1 = lambda0 - f/f_der;
    dif = abs(lambda1 - lambda0);
    lambda0 = lambda1;

end

% IS IT WORKING?
%numerator = sum( abs(z).^2 / (iGamma + lambda0).^2 );
%denominator = (sum( abs(z).^2 / (iGamma + lambda0) )).^2;
%f = numerator/denominator - rho

if (lambda0 < lblambda) | (lambda0 > ublambda),
    lambda0 = NaN;
end

% ---------------------------------------------------------------
```