

PhD Thesis
February 2013

THEORY AND APPLICATIONS OF
BOUNDED COMPONENT ANALYSIS IN
COMPLEX-VALUED SIGNAL PROCESSING



Author: Pablo Aguilera Bonet

Tutor: Sergio A. Cruces Álvarez

*Department of Signal Theory and Communications
University of Seville*

Dr. **Sergio Antonio Cruces Álvarez**, Profesor Titular del Departamento de Teoría de la Señal y Comunicaciones (Escuela Técnica Superior de Ingenieros, Universidad de Sevilla).

HACE CONSTAR:

Que el trabajo titulado "*Theory and Applications of Bounded Component Analysis in Complex-Valued Signal Processing*", presentado por D. Pablo Aguilera Bonet para optar al título de Doctor Ingeniero de Telecomunicación, ha sido realizado en el departamento de Teoría de la Señal y Comunicaciones de la Universidad de Sevilla bajo su supervisión, y que reúne todas las condiciones exigidas a los trabajos de doctorado.

Sevilla, Febrero de 2013

Sergio A. Cruces Álvarez

The work of D. Pablo Aguilera Bonet has been supported by the Spanish National Board of Scientific and Technological Research (CICYT) under Projects TEC2008-06259 and TEC2011-23559.

A mis padres.

A Elena.

AGRADECIMIENTOS

Esta Tesis Doctoral no habría sido posible sin el apoyo de Sergio A. Cruces. Sus consejos y su guía fueron fundamentales para el despegue de mi carrera investigadora. También me he sentido plenamente respaldado por Carlos, Javier, Iván, María José, Uxi y Juan Antonio del Grupo de Sistemas de Radiocomunicación.

Desde mi primer día en el Departamento de Teoría de la Señal, me sentí gratamente integrado gracias a la cercanía de Juanjo, Rafa, Javier, Irene, Susana, Eva, José Ramón, Aurora, Fran, Foncu y el resto de compañeros. El buen ambiente fue un factor decisivo para decidirme a pasar estos fantásticos años en la sala al final del pasillo.

Mención especial merecen en mi corazón los que ya considero amigos para toda la vida, Pablo, Michel y Luis, aunque esta se empeñase algún día en mandarnos a cada uno a una punta del planeta.

Durante los meses que pasé en Londres, el frío invierno no lo fue tanto gracias a la simpatía del profesor Danilo P. Mandic y su equipo en el Department of Electrical and Electronic Engineering del Imperial College.

También me acuerdo en estos momentos de mis compañeros de carrera como Ale, Alfonso y Álvaro; así como de mis compañeros de aventuras y risas en tantas y tantas ocasiones en Jerez: Dani, Abel, Galván, Alex, Manuel Jesús, Arturo, Víctor, Jorge, Carlos, Calidad, Gonzalo, Iñaki, Andrés, Kuko, Elena, Emma, Sara, Isa, Alba y los demás.

Gracias por transmitirme cada fin de semana su alegría a Maria Luisa, Paco y Marisa, y por tratarme como si estuviese en mi casa.

A mi familia debo agradecerle todo. Gracias a mis abuelos Isabel, Alfonso, Gloria y Ernesto por alimentar mi amor por la vida y por la ciencia. Gracias también a mis tíos y tías, así como a mi primo y a mis primas, por sacarme siempre una sonrisa cuando lo necesitaba. Gracias a Irene por su cercanía y sus consejos al haber recorrido este camino antes que yo. Gracias infinitas a mis padres, Pepe y Gloria, a los que debo todo lo que soy.

Y gracias a Elena, la mejor compañera que podría haberme tocado para el viaje de la vida, por estar a mi lado en los buenos y en los malos momentos.

Any sufficiently advanced technology is indistinguishable from magic.

- A. C. Clarke -

ABSTRACT

This Thesis is dedicated to the presentation, study and application of Bounded Component Analysis (BCA), a latent component analysis technique which can be applied in different manners to solve a wide spectrum of complex-valued signal processing problems.

An introduction to the theory of complex calculus is presented, where the concepts of cost functions, Wirtinger derivatives, and gradients are illustrated in order to show the methodology to solve optimization problems where there are complex-valued signals and systems involved. Also, we conduct an overview of the recent advances in the field of augmented statistics and widely linear modeling.

The theory of BCA is deeply reviewed. It is based upon geometrical properties of the sequences in the complex domain, which connects two infrequently related fields such as convex geometry and time series analysis. A criterion based upon the convex perimeter functional is derived for both blind and semi-blind scenarios. This eventually allows the extraction of one or more sources from an unknown mixture, as long as they satisfy the main hypothesis of BCA. Furthermore, the sources can be dependent, thus separating them in scenarios where Principal Component Analysis (PCA) and Independent Component Analysis (ICA) fail.

We also present the convergence study of a BCA extraction method based upon the gradient descent algorithm, with the proposal of several values for the step size, each one with its own advantages and drawbacks. On one hand, we propose fast step sizes based upon the local analysis of the shape of the cost function, although they do not guarantee the stability of the iterations. On the other hand, we recommend a step size that guarantees the global monotone convergence of the iterations to local minima, derived from a comprehensive study of the geometry of the update equations.

In addition, the study of under-determined and noisy mixtures is addressed. We show how the application of the supervised version of BCA to these scenarios can be used as a pre-processing step for a further refined classification in communications systems. This two stage recovery methodology allows us to successfully separate bounded sources where other well-known criteria like Minimum Mean Square Error (MMSE) fails.

RESUMEN

Esta Tesis Doctoral está dedicada al estudio de la teoría y de las aplicaciones del Análisis de Componentes Acotadas (BCA), una técnica de análisis de componentes latentes cuya aplicación es resolver un buen número de problemas que surgen al tratar con señales de variable compleja.

Al principio de la Tesis Doctoral, se introducen algunos conceptos importantes relacionados con el cálculo con variable compleja, tales como funciones de coste, derivadas de Wirtinger y gradientes. Gracias a ello, se presentan algunas aplicaciones de cómo se pueden resolver problemas clásicos de tratamiento de señal con variable compleja. Además, hacemos un repaso a los últimos avances en el campo de la estadística aumentada y el modelado widely linear.

Se revisa en profundidad la teoría que sustenta BCA. Básicamente, está basada en las propiedades geométricas de las secuencias cuando se representan en el plano complejo, lo cual conecta dos campos de la matemática inusualmente relacionados, tales como la geometría convexa y el análisis de series temporales. Tras esta revisión a fondo, se presentan criterios basados en el perímetro convexo de un vector, para el caso ciego y también para el caso semi-ciego. Así, podemos extraer una o más fuentes de una mezcla lineal, siempre que estas cumplan las hipótesis básicas de BCA. Es más, las fuentes pueden ser dependientes, consiguiendo una separación correcta en escenarios donde el Análisis de Componentes Principales (PCA) o el Análisis de Componentes Independientes (ICA) fallan.

Además, presentamos el estudio de convergencia de un método de extracción de fuentes basado en BCA, e implementado mediante un algoritmo de descenso por gradiente. Es por esto que podemos presentar toda una serie de valores para el paso adaptativo, cada uno de ellos con sus propias ventajas e inconvenientes. Por una parte, proponemos algunos pasos rápidos basados en el análisis del comportamiento local de la función de coste, pero que no garantizan la estabilidad de las iteraciones. Por otro lado, recomendamos un paso que garantiza la convergencia global monótona de las iteraciones a un mínimo local, aunque no es tan rápido como los anteriores. Este análisis está basado en un estudio detallado de la geometría de la ecuación de actualización.

Finalmente, se aborda el problema de extracción supervisada de fuentes en mezclas sub-determinadas, en presencia de ruido. Mostramos cómo el modelo semi-ciego de BCA puede utilizarse como un paso previo a un método de clasificación no-lineal, con el objetivo de recuperar señales acotadas cuando no tenemos suficientes grados de libertad en el sistema. Esto es algo que otros métodos bien conocidos como el Mínimo Error Cuadrático Medio (MMSE) no son capaces de resolver.

CONTENTS

Agradecimientos	7
Abstract	11
Resumen.....	13
Contents	15
1 Introduction.....	23
1.1 Motivation.....	24
1.2 Aims and Scope	25
1.3 Thesis Overview	26

2	Complex Calculus	29
2.1	Introduction	29
2.2	Complex Functions and Variables	30
2.3	Derivatives of Complex Functions.....	31
2.3.1	Holomorphic functions.....	31
	<i>Cauchy-Riemann conditions</i>	32
	<i>Structure of holomorphic functions</i>	34
2.3.2	Non-holomorphic functions: $\mathbb{C}\mathbb{R}$ calculus.....	36
	<i>Wirtinger derivatives</i>	37
	<i>The Cauchy-Riemann condition revisited</i>	39
	<i>Rules of complex derivation</i>	40
	<i>Multivariate $\mathbb{C}\mathbb{R}$ calculus</i>	41
2.4	Optimization with Complex Variables.....	43
2.4.1	Stationary points	43
2.4.2	The direction of maximum rate of change.....	44
2.4.3	Function minimization via a steepest descent algorithm	46
2.5	Augmented Complex Descriptions.....	48
2.5.1	Properness and circularity	48
2.5.2	Connection between real and complex descriptions	50
	<i>Widely linear transformations</i>	51
2.5.3	Second-order augmented statistics	52
2.6	Conclusions.....	56

3	Complex-Valued Signal Processing.....	59
3.1	Introduction	60
3.2	The Minimum Mean Square Error Criterion.....	60
3.2.1	Geometrical interpretation of the MMSE criterion.....	64
3.3	The Complex Least Mean Squares Algorithm	65
3.3.1	Convergence of the CLMS algorithm.....	66
3.4	Blind Source Separation and Blind Source Extraction in the Complex Domain.....	70
3.4.1	Principal Component Analysis, decorrelation and whitening ..	72
3.4.2	Independent Component Analysis.....	74
3.5	Widely Linear Wiener Filtering.....	78
3.6	The Strong Uncorrelating Transform (SUT).....	84
3.7	Conclusions.....	86
4	Bounded Component Analysis.....	89
4.1	Introduction	90
4.2	Signal Model and Notation.....	91
4.2.1	Blind Signal Separation and Extraction.....	91
	<i>Blind Source Extraction</i>	92
	<i>Blind Source Separation</i>	93
4.2.2	Theory of convex bodies	93
4.3	Main Assumptions of Bounded Component Analysis.....	97

4.4	Identifiability, Separability, and Indeterminacies.....	100
4.5	The Convex Perimeter of the Output.....	102
4.6	The Criterion of Minimum Normalized Convex Perimeter 106	
4.6.1	Properties of the normalized convex perimeter.....	107
4.6.2	Consistency of the real case.....	111
4.7	Optimization of the Cost Function.....	112
4.7.1	Whitening as a pre-processing step.....	112
4.7.2	Differentiability of the cost function.....	113
4.7.3	A gradient descent algorithm.....	114
4.7.4	Practical computation of the convex perimeter.....	115
4.7.5	The gradient of the normalized convex perimeter.....	117
4.8	Blind Source Extraction and Separation by using BCA.....	118
4.8.1	A BCA algorithm for the Blind Source Extraction.....	118
	<i>Computer simulations</i>	119
4.8.2	Computational complexity.....	120
4.8.3	Blind Source Separation.....	121
	<i>Computer simulations</i>	123
4.9	Latent Component Decomposition by using BCA.....	125
4.9.1	Linear model.....	126
4.9.2	The sequential recovery of the components.....	127
4.9.3	Computer simulations.....	129
4.10	Conclusions.....	131

5 Convergence Study of a BCA Extraction

Algorithm	133
5.1 Introduction	134
5.2 The Special Shape of the Cost Function.....	134
5.3 Step Sizes Based On the Shape of the Cost Function.....	136
5.3.1 The Newton-Raphson step size	137
5.3.2 A step size that uses the second-order information	139
5.4 A Step Size that Guarantees the Stability	143
5.4.1 The underlying structure of the iteration.....	143
5.4.2 A convenient parameterization	149
<i>The global monotonous convergence condition</i>	152
5.4.3 Stable bounds for the step size.....	153
<i>A step size that prevents the change of the $\boldsymbol{\pi}(\mathbf{2})$ coefficient: $\boldsymbol{\mu}_P$</i>	155
<i>A step size that guarantees the monotonous convergence: $\boldsymbol{\mu}_S$</i>	156
<i>A step size that prevents the leakage from the current basin of the cost function: $\boldsymbol{\mu}_B$</i>	158
5.4.4 Practical estimators: the proposed step size	159
<i>Evaluating the norms of the gradients</i>	159
<i>Bounds for the non-negative normalized coefficients</i>	160
<i>A practical estimator for the $\boldsymbol{\mu}_P$ step size</i>	163
<i>A practical estimator for the $\boldsymbol{\mu}_S$ step size</i>	163
5.5 Computer simulations.....	164
5.6 Conclusions.....	168

6	Bounded Component Analysis of the Training Error.....	171
6.1	Introduction	172
6.2	Signal Model.....	173
6.3	Under-determined Mixtures.....	175
6.4	Bounded Component Analysis of the Training Error	176
6.4.1	The convex perimeter functional as a cost function.....	177
6.4.2	Structure of the convex perimeter.....	178
6.5	The Supervised BCA Extraction Algorithm	179
6.5.1	The gradient of the convex perimeter	179
6.5.2	The step size	180
	<i>The Newton-Raphson step size.....</i>	<i>180</i>
	<i>A step size that uses the second order information.....</i>	<i>181</i>
6.6	Structural Similitudes and Differences with the Mean Square Error	183
6.7	Geometry of the Norms	185
6.7.1	The 1-norm	185
6.7.2	The 2-norm	187
6.7.3	Geometry of the norms in under-determined problems	188
6.8	A Partial Zero-Forcing Criterion.....	190
6.8.1	Constraints on the optimization problem.....	190
6.8.2	The exact Partial-Zero Forcing solution.....	191

6.8.3	The gradient of the convex perimeter with PZF.....	193
6.9	Computer Simulations	195
6.9.1	Signal extraction in an under-determined mixture.....	195
	<i>The extended vector of coefficients.....</i>	<i>196</i>
	<i>The Performance Index.....</i>	<i>199</i>
	<i>The output signal</i>	<i>201</i>
6.9.2	Use as a pre-processing step for classification.....	202
	<i>Implementing a hard decisor based on the Maximum A Posteriori rule.....</i>	<i>204</i>
	<i>Implementing a Support Vector Machine.....</i>	<i>208</i>
6.10	Conclusions.....	208
7	Conclusions	211
8	Open Research Lines.....	215
	References	217
	Index of Figures	227
	Index of Terms	235

1 INTRODUCTION

We live in a society exquisitely dependent on science and technology, in which hardly anyone knows anything about science and technology.

- C. Sagan -

This Thesis presents in an unified manner the theory and applications of a novel latent component analysis technique. It goes along with multiple examples of scenarios where both classical and new problems in complex-valued signal processing are solved.

The extraction of one or more latent sources from a set of observations as a result of a linear mixture is a problem that, due to its simplicity on the description, has found many applications in diverse topics. These applications feed some of the most growing research fields in the science and technology, such as communications systems with multiple antennas, speech, image and signal processing, and biomedical engineering with multiple sensors.

1.1 Motivation

Numerous works has been published over the last decades, providing feasible solutions to the problem of Blind Signal Separation (BSS) and Extraction (BSE), like those based on the decorrelation of sources in Principal Component Analysis (PCA) (Jolliffe, 2002), on their statistical independence and non-Gaussianity in Independent Component Analysis (ICA) (Hyvärinen, 2001), or on their non-negativity in Non-negative Matrix Factorization (NMF) (Cichocki, 2009).

Among the wide spectrum of existing ICA methods for the *blind* recovery of the sources, some of them do rely also on geometric properties of the sources, like their finite cardinality (Gamboa, 1997), their statistical range (Pham, 2000), the measure of their support set (Erdogan, 2006), the shape of the constellation (Mansour, 2002), or their known alphabet (Zarzoso, 2006).

However, there are also *supervised* or semi-blind methods that use a small amount of samples from a desired source, using it as a reference in addition to the structural properties of the components of the mixture. Well-known solutions derived from this approach are the Minimum Mean Square Error (MMSE) (Adali, 2010) or the Zero Forcing (ZF) methods (Johnson Jr, 1995).

These blind, semi-blind and supervised criteria that solve the decomposition of the observations into components, are usually mathematically expressed by means of a *cost function* or a *contrast function*. Those functions proportionate a metric for the different one-dimensional projections of the observations, based upon some known hypothesis about the components and upon the invertibility of the mixture.

The most reviewed instantaneous mixtures are those that have more observations than sources (*over-determined* systems), or at least the same quantity (*determined* systems). The columns of the mixing matrix form a complete or over-complete basis for the space, which in a noiseless situation eventually results in the perfect extraction of a source. More recently, the study of *under-determined* systems, both in the blind and the supervised case, has been also addressed, like in (Comon, 2004), (Theis, 2004) or (Donoho, 2006), among others.

Nevertheless, the majority of the works in the field of signal separation are focused on the real-valued case. During the past decades, the study of real cost function of *complex variables* has experimented a growth, with key contributions gathered in books like (Schreier, 2010) and (Mandic, 2007). The use of complex-valued signals and mixtures allows us to model the richer behaviors that are found on real life applications. However, the extension of signal processing

concepts to the field of complex random processes is not as straightforward as one could a priori think.

1.2 Aims and Scope

With the realization of this Thesis, we aim to contribute to the improvement of the source extraction, separation and decomposition methods, by means of the study of a novel technique, which combine *time series analysis* and *convex geometry*. The developed methods must be adaptive to real applications, aiming to exploit the a priori known information about the problem.

The main contributions are concreted on the following lines of research:

- A deep review of the *Bounded Component Analysis (BCA)* of linear mixtures, a framework developed in (Cruces, 2010), which estimates the latent components of the observations by means of the property of compactness of the finite support of the mixed signals. We will see that one can dispense with the mutual statistical independence hypothesis of the transmitted signals.
- The blind decomposition of a set of observations into bounded latent components (Cruces, 2010), with the proposal of an iterative *gradient descent algorithm* entirely based on the minimization of the *convex perimeter* of the estimate and the deflation of the extracted component from the original set of observations (Aguilera, 2012).
- The conditions for the separability of the sources in a blind scenario are very important to guarantee a successful extraction, separation or decomposition. Also, we conduct a comprehensive study of the shape of the cost function, which is free of erroneous *minima* under some given hypotheses. We aim to propose recommendations for the *step size* depending on the desired behavior of the algorithm, such as the rapid *convergence* (Aguilera, 2012) or the guarantee of the stability in a given sense (Aguilera, 2013).
- The application of new methods to solve supervised or semi-blind problems, like in (Aguilera-Bonet, 2012) and (Cruces, 2011), where there is a small sequence of pilot symbols that can be transmitted to train the receiver of a communication system. This has a direct application on the channel compensation of multi-antenna or multi-user wireless systems; where the particularities of the physical medium produces interference on the signal form the desired user.
- The solution to noisy under-determined linear mixtures, where the

number of sources is smaller than the number of observations, which limits the exact recovery of the desired source. The proposed solution to these problems can be seen as a two-block system that uses BCA as a pre-processing stage for a *non-linear* refinement of the estimation.

- The study of the *widely linear models* to describe the more complicated nature of the mixtures and the sources in real applications, as well as the novel tools that *augmented statistics* provide to complex-valued signal processing.

Secondary tasks should be completed before achieving the described goals. First of all, the complex calculus with non-holomorphic variables must be presented. Several examples and the methodology used to derive the gradients of the cost functions is clarified in the solution of practical problems presented in complex-valued signal processing. Some of these well-known solutions are the reference to beat with our proposed algorithms in this Thesis.

Among our priorities, we intend to present in a unified manner the theory and the applications of Bounded Component Analysis to the solution of complex-valued signal processing problems, like signal extraction, separation and latent component decomposition.

1.3 Thesis Overview

The above aims of this Thesis will be developed through seven chapters, starting with this introduction in which the motivation and the objectives are discussed.

In Chapter 2, we set the fundamentals of the differential calculus with complex-variables. We review concepts like holomorphic functions, Wirtinger calculus, and gradient optimization in the complex domain. In addition, the last trends in complex-valued signal processing are presented: the use of widely linear models and augmented statistics.

In Chapter 3, we review some of the most successful although simple applications: the complex Wiener Filter, the Complex Least Mean Squares (CLMS) algorithm, and blind techniques like PCA and a simple kurtosis-based ICA. This chapter set the basis for the methodology when dealing with real-valued functions of complex variable. During the rest of the Thesis, we compare the differences and similitudes of our proposed solutions with these well-known signal processing methods. Also, we review briefly some direct applications from the theory of non-circular random processes.

In Chapter 4, the theory of Bounded Component Analysis is presented, with

emphasis on the comprehension of the hypothesis, the conditions for the separability of the sources and the introduction of an appropriate cost function like the convex perimeter. We also present some applications of the firsts algorithms derived from BCA: blind source extraction, separation and component decomposition, with their correspondent computer simulations.

In Chapter 5, the convergence analysis of BCA extraction algorithm is presented. Within this study, several step sizes for the optimization algorithm were derived, each one with its own benefits and drawbacks. We propose a convenient step size that guarantees the global monotonous convergence of the algorithm in a given sense, and provide some computer simulations to corroborate the theoretical results.

In Chapter 6, a semi-blind model for BCA is presented. This time, we propose a small training sequence to adapt the extraction vector to recover the desired source. We introduce the training error sequence, whose convex perimeter is minimized in the pursuit of a solution. The model is completed with the addition of noise, which forces us to the modification of the optimization criterion. Computer simulations show that the proposed solution beats in performance other classical solutions (when the mixture is under-determined).

Chapter 7 gives a final overview of the whole Thesis, and provides some ideas for further research. Some of these ideas are currently under development.

2 COMPLEX CALCULUS

The complex analytic function turns out to be much more special, enjoying many beautiful properties not shared by the run-of-the-mill function from ordinary real calculus.

- F. Flanigan -

The use of complex numbers and complex variables in signal theory is not a new topic. Classical papers that address this issue are, for example, (Widrow, 1985) and (Mooers, 1973). However, during the last decade, a new interest on the field of complex numbers has been observed, motivated by new applications such as biomedical signal processing (Javidi, 2010), communications (Schreier, 2003), and array processing (Delmas, 2004).

This chapter is mainly based upon the introduction (Kreutz-Delgado, 2006), and the books (Mandic, 2009), (Schreier, 2010), and (Hjørungnes, 2011).

2.1 Introduction

In this chapter, the main purpose is to provide the reader a comprehensive overview of the tools that researchers have developed to treat with the more

sophisticated behaviors of complex signals. One could initially think that the signal processing in the complex domain reduces to the systematic use of the conjugate operator on products, but we will soon realize that the structure underlying in the field of complex functions is richer than in the field of real functions.

The chapter is organized as follows. In Section 2.2 we present some basic definitions for complex functions and variables. In Section 2.3, we justify a classification of the functions of complex-variable, and we present the theory for finding the derivatives of such functions. In Section 2.4, we present some concepts that are good to know when minimizing or maximizing real functions of complex variables like, e.g., where are the stationary points, which is the direction of maximum rate of change, etc. In Section 2.5, we present the approach called augmented statistics, a recent point of view for the processing of complex-valued signals that exploits further information. Section 2.6 summarizes the conclusions to this chapter.

2.2 Complex Functions and Variables

In this section we present the notation of complex variables and functions, and some of their basic operators and properties.

The complex argument of a function can be a scalar $z \in \mathbb{C}$, a vector $\mathbf{z} \in \mathbb{C}^n$, or a matrix $\mathbf{Z} \in \mathbb{C}^{n \times m}$. We will focus on the scalar case to present some key concepts in the theory of complex-valued functions and variables. If we call x and y to the real and imaginary parts of z , respectively, then

$$z = x + jy = \Re\{z\} + j\Im\{z\}, \quad (2.1)$$

where j represents the imaginary unit so $j^2 = -1$. Operators $\Re\{\cdot\}$ and $\Im\{\cdot\}$ will return, the *real* and *imaginary* parts of their argument, respectively. The *absolute value* of a complex number z is $|z|$ and is computed as

$$|z| = \sqrt{zz^*} = \sqrt{x^2 + y^2}. \quad (2.2)$$

The *conjugate* operator $(\cdot)^*$ changes the sign of the imaginary part of its argument,

$$z^* = x - jy = \Re\{z\} - j\Im\{z\}. \quad (2.3)$$

Although this operation may look very simple, it is highly useful when dealing with complex calculus. For example, we can express real and imaginary parts as a linear combination of a complex variable and its conjugate,

$$x = \Re\{z\} = \frac{1}{2}(z + z^*), \quad (2.4)$$

$$y = \Im\{z\} = \frac{1}{2j}(z - z^*). \quad (2.5)$$

We introduce now the concept of *complex functions*. A complex-valued function is a mapping from a given *domain* D into the complex scalar ($f \in \mathbb{C}$), vector ($\mathbf{f} \in \mathbb{C}^n$), or matrix ($\mathbf{F} \in \mathbb{C}^{n \times m}$) domain. For them, one can define the same concepts seen before (real and imaginary parts, absolute value, and conjugate). Hereinafter, we will focus on the scalar case.

2.3 Derivatives of Complex Functions

The main problem when working with complex variables is the lack of ordering. We cannot say that a given complex number is greater or lower than another.

Many applications on engineering require the optimization of the so called *cost functions*, *objective functions*, or *loss functions*. These are functions that, when maximized or minimized, lead to specific solutions of a certain criterion. Practical cost functions must be, by definition, real-valued. This is due to the fact that we are interested in ordered values, so we can wander on the surface of the function to maximize or minimize its real value. Examples of scalar and real-valued functions widely used in engineering are the temperature or the electric potential.

2.3.1 Holomorphic functions

When dealing with complex variables, the notion of derivative is not as direct and intuitive as in the real variable case. Usually, traditional courses on complex variable calculus start with the concept of *holomorphic function*.

Definition (Holomorphic function): Let $D \subseteq \mathbb{C}$ be the domain of the scalar function $f : D \rightarrow \mathbb{C}$. Then, $f(z)$ is an holomorphic function in the domain D if the limit

$$f'(z) = \lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} \quad (2.6)$$

exists for all $z \in D$.

Cauchy-Riemann conditions

For a function to be holomorphic, the previous limit (2.6) must be independent of the direction which f approaches zero in the complex plane. This, although can be seen as a minor issue, is indeed a very strong condition imposed on the function $f(z)$.

Example 2.1 Suppose that our function $f(z): \mathbb{C} \rightarrow \mathbb{C}$ is the square of the Euclidean distance, defined as

$$f(z) = |z|^2 = zz^*. \quad (2.7)$$

We want to compute the limit (2.6) to obtain the complex derivative of $f(z)$. However, we see that this limit takes different values depending on the approximation path to the point z . Substituting into the limit,

$$\begin{aligned} f'(z) &= \lim_{\Delta z \rightarrow 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} = \lim_{\Delta z \rightarrow 0} \frac{|z + \Delta z|^2 - |z|^2}{\Delta z} \\ &= \lim_{\Delta z \rightarrow 0} \frac{(z + \Delta z)(z^* + \Delta z^*) - zz^*}{\Delta z} \\ &= \lim_{\Delta z \rightarrow 0} \frac{(\Delta z)z^* + z(\Delta z^*) + \Delta z(\Delta z^*)}{\Delta z}. \end{aligned} \quad (2.8)$$

We can see that it does not exist, because it depends on the path of approximation to the point where it is evaluated. For example, let us take the two simplest cases of approximation over the coordinate axes, with $\Delta z = \Delta x + j\Delta y \rightarrow 0$:

- Case 1: $\Delta y = 0$, while $\Delta x \rightarrow 0$:

$$\begin{aligned} f'(z) &= \lim_{\Delta x \rightarrow 0} \frac{(\Delta x)z^* + z(\Delta x) + (\Delta x)^2}{\Delta x} \\ &= \lim_{\Delta x \rightarrow 0} (z^* + z + \Delta x) = z^* + z = 2\Re\{z\}. \end{aligned} \quad (2.9)$$

- Case 2: $\Delta x = 0$, while $\Delta y \rightarrow 0$:

$$\begin{aligned} f'(z) &= \lim_{\Delta y \rightarrow 0} \frac{j(\Delta y)z^* - jz(\Delta y) + (\Delta y)^2}{\Delta y} \\ &= \lim_{\Delta y \rightarrow 0} (j(z^* - z) + \Delta y) = j(z^* - z) = -2\Im\{z\}. \end{aligned} \quad (2.10)$$

And it is evident that $2\Re\{z\} \neq -2\Im\{z\}$, so the derivative $f'(z)$ does not exist.

We conclude that some functions are non-holomorphic, because the limit (2.6) does not always exist. The following theorem establishes sufficient and necessary conditions for the function to be holomorphic.

Theorem 2.1 (Cauchy-Riemann conditions)

Let $f(z) = u(x, y) + jv(x, y)$ be a complex function of complex variable. For $f(z)$ to be holomorphic, the real and imaginary parts of f and z must satisfy the Cauchy-Riemann conditions:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}. \quad (2.11)$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}. \quad (2.12)$$

Proof: (Necessity) For a function to be holomorphic, it must satisfy (2.6) independently of the path of approximation to the point z when $\Delta z = \Delta x + j\Delta y \rightarrow 0$. If we expand (2.6) in real and imaginary parts of z , and of f :

$$f'(z) = \lim_{\substack{\Delta x \rightarrow 0 \\ \Delta y \rightarrow 0}} \left(\frac{u(x + \Delta x, y + \Delta y) + jv(x + \Delta x, y + \Delta y) - u(x, y) - jv(x, y)}{\Delta x + j\Delta y} \right). \quad (2.13)$$

Let us consider now the two simplest cases for the approach of $\Delta x \rightarrow 0$ and $\Delta y \rightarrow 0$, that correspond to the coordinate axes:

- Case 1: $\Delta y = 0$, while $\Delta x \rightarrow 0$:

$$\begin{aligned} f'(z) &= \lim_{\Delta x \rightarrow 0} \left(\frac{u(x + \Delta x, y) + jv(x + \Delta x, y) - u(x, y) - jv(x, y)}{\Delta x} \right) \\ &= \lim_{\Delta x \rightarrow 0} \left(\frac{u(x + \Delta x, y) - u(x, y)}{\Delta x} + j \frac{v(x + \Delta x, y) - v(x, y)}{\Delta x} \right) \\ &= \frac{\partial u(x, y)}{\partial x} + j \frac{\partial v(x, y)}{\partial x}. \end{aligned} \quad (2.14)$$

- Case 2: $\Delta x = 0$, while $\Delta y \rightarrow 0$:

$$\begin{aligned}
 f'(z) &= \lim_{\Delta y \rightarrow 0} \left(\frac{u(x, y + \Delta y) + jv(x, y + \Delta y) - u(x, y) - jv(x, y)}{j\Delta y} \right) \\
 &= \lim_{\Delta y \rightarrow 0} \left(\frac{u(x, y + \Delta y) - u(x, y)}{j\Delta y} + j \frac{v(x, y + \Delta y) - v(x, y)}{j\Delta y} \right) \\
 &= \frac{\partial v(x, y)}{\partial y} - j \frac{\partial u(x, y)}{\partial y}.
 \end{aligned} \tag{2.15}$$

To ensure uniqueness of the limit (2.13), equation (2.14) must be equal to equation (2.15). Identifying real and imaginary parts, we get the Cauchy-Riemann conditions.

■

Structure of holomorphic functions

Let us take a look on the properties of holomorphic functions. Although the notation that it is used in complex calculus is very similar to the one used in real calculus, an holomorphic function $f(z)$ has a certain structure that makes itself somewhat special. Specifically, the following results are equivalent:

- The derivative $f'(z)$ exists and is continuous.
- The function $f(z)$ is holomorphic (that is, *analytic*¹ in z).
- The function $f(z)$ satisfies the Cauchy-Riemann conditions.
- All the derivatives of the function $f(z)$ exist, and $f(z)$ admits convergent power series expansion.
- Real $u(x, y)$ and imaginary $v(x, y)$ parts of the function $f(z)$ are harmonic functions, that is, they satisfy Laplace equations:

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = 0, \tag{2.16}$$

$$\frac{\partial^2 v(x, y)}{\partial x^2} + \frac{\partial^2 v(x, y)}{\partial y^2} = 0. \tag{2.17}$$

¹ A function is analytic in a domain if it admits convergent power series expansion in such domain. That implies that the function has derivatives of all orders. For a complex function of complex variable, the term *analytic* has been recently substituted by the term *holomorphic*, although both are synonyms and we could interchange them. Specifically, we can say that a function of real variable that admits real power series expansions is analytic (*real analytic*), while a function of complex variable that admits complex power series expansion is holomorphic (*complex analytic*).

It is clear that, when a function is holomorphic, we are imposing a big structure and strong properties on it. A holomorphic function can also be known as *complex differentiable*, *complex analytic* or *regular*.

Although many important complex functions are holomorphic and thus complex differentiable, unavoidably we are going to find lots of functions that are not.

Example 2.2 *Some common functions that are not holomorphic:*

- The function $f(z) = z^*$, that represent the complex conjugate of z , does not satisfy the Cauchy-Riemann conditions.
- Functions $\Re(z) = (z + z^*)/2$ and $\Im(z) = (z - z^*)/2j$. Neither of them do satisfy it.
- The function $f(z) = |z|^2 = zz^* = x^2 + y^2$ is not holomorphic.

Let us realize that all the functions presented in the example are non-holomorphic, and they can be expressed as a function of the independent variable z and its complex conjugate z^* . This is a simple test to identify whether a function is holomorphic or not.

It seems that there are not many functions in the signal processing field that satisfies the conditions to be holomorphic and complex differentiable. The following theorem explains why.

Theorem 2.2 (The real-valued holomorphic function)

Let $D \subseteq \mathbb{C}$ be a domain in the complex plane, and let $f(z) : D \subseteq \mathbb{C}$ be a real holomorphic function. Then, $f(z)$ must be the constant function, for all z .

Proof: If $f(z)$ takes only real values, necessarily $v(x, y) = \Im(z) = 0$. Then, if $f(z)$ is holomorphic, it must satisfy the Cauchy-Riemann equations,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} = 0, \quad (2.18)$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y} = 0. \quad (2.19)$$

So the real part $u(x, y)$ must be constant throughout the z plane, that is,

$$f(z) = \text{const.}, \forall z. \quad (2.20)$$



This is a classical result that reduces the set of real holomorphic functions to only the constant function. In practice, cost functions (as we stated at the start of Section 2.3) are real but necessarily non-constant, so they are not holomorphic functions, and their study cannot be done by using classical tools for complex variables.

Let us remark this fact: if we look for an optimization procedure to find the optimal point of a real, non-constant cost function, we find that the function is not holomorphic. Thus, its derivative respect the independent complex variable z does not exists in the conventional sense.

For all these reasons, it is necessary an alternate formulation for the calculus of derivatives of real functions with complex variables and, in general, non-holomorphic functions.

2.3.2 Non-holomorphic functions: $\mathbb{C}\mathbb{R}$ calculus

We want to find the derivative of a non-holomorphic, real-valued function $f \in \mathbb{R}$ with respect to a complex variable that can be a scalar z , a vector \mathbf{z} or a matrix \mathbf{Z} . To simplify the notation, we will start on the scalar case. Initially, one could adopt either of the two following points of view.

- The first one consists on rewriting f depending on the real x and imaginary y parts of the complex variable z . Once this is done, we can find the derivatives of the function, reordered with respect to those two independent real variables. We can indeed treat f as a real function with real variables x and y :

$$f(z) \equiv f(x, y): \mathbb{R}^2 \rightarrow \mathbb{R}. \quad (2.21)$$

Using this trick, we see that the existence of $\partial f / \partial x$ and $\partial f / \partial y$ is sufficient for f to be differentiable. The key idea is to exploit the structure of the vector space \mathbb{R}^2 underlying in \mathbb{C} when computing the derivatives.

- The second option, much more stylish and with advantages with respect the first one, consists on treating the variables z and z^* independently. We are not saying that a complex variable z and its complex conjugate z^* are independent, but that *their cross-derivatives are zero*. Sometimes, the notation $f(z, z^*)$ is forced to remark this fact. This notation was proposed for the first time in (Brandwood, 1983).

Those new derivatives, $\partial f / \partial z$ and $\partial f / \partial z^*$, are very useful when dealing with non-holomorphic functions. We call them the *real derivative* (derivative in \mathbb{R}) and the *real conjugate derivative* (derivative in \mathbb{R}^*). That is to highlight the

difference from the *complex derivative* (*derivative in \mathbb{C}*), that as we have seen in Section 2.3.1, only works on holomorphic functions.

To be consistent, this new approach should reduce to the classic complex derivative presented in (2.6). This introduces the so-called *Wirtinger calculus* or $\mathbb{C}\mathbb{R}$ *calculus* (Kreutz–Delgado, 2006).

Wirtinger derivatives

The pair of variables (z, z^*) defines a dual representation of the same plane that the pair of real variables (x, y) .

In general, we can define the pair of partial real derivatives for a function of complex variable $f(z)$ as the following.

Definition (Wirtinger derivatives): Let $f(z)$ be a scalar function of complex variable z . Then, we can define the following formal² partial derivatives:

- \mathbb{R} derivative of $f(z, z^*) \triangleq \left. \frac{\partial f(z, z^*)}{\partial z} \right|_{z^* = \text{const.}}$.
- \mathbb{R}^* derivative of $f(z, z^*) \triangleq \left. \frac{\partial f(z, z^*)}{\partial z^*} \right|_{z = \text{const.}}$.

Those derivatives are related with the real and imaginary parts derivatives in the following way.

Theorem 2.3 (Relation between Wirtinger and real derivatives)

Let $f(z)$ be a scalar function of complex variable z . Then, its \mathbb{R} and \mathbb{R}^* derivatives can be expressed as:

$$\left. \frac{\partial f(z, z^*)}{\partial z} \right|_{z^* = \text{const.}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) \quad (2.22)$$

$$\left. \frac{\partial f(z, z^*)}{\partial z^*} \right|_{z = \text{const.}} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) \quad (2.23)$$

² Note that these derivatives are purely *formal* definitions. Actually, we cannot vary z without altering z^* , and vice versa. However, z and z^* are independent in the sense that $\partial z / \partial z^* = \partial z^* / \partial z = 0$, as we see in Example 2.3.

Proof: Let us express the differential of the function of two real variables $f(x, y)$:

$$df(x, y) = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy. \quad (2.24)$$

As $f(x, y) = u(x, y) + jv(x, y)$, we can expand using the chain rule of derivation:

$$df(x, y) = \frac{\partial u}{\partial x} dx + j \frac{\partial v}{\partial x} dx + \frac{\partial u}{\partial y} dy + j \frac{\partial v}{\partial y} dy. \quad (2.25)$$

Now, using (2.4) and (2.5) we make the following changes of variable over the differentials of the real ($dx = (dz + dz^*)/2$) and imaginary parts ($dy = (dz - dz^*)/2j$). After reordering, we obtain

$$df = \frac{1}{2} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + j \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) \right) dz + \frac{1}{2} \left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} + j \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right) dz^*. \quad (2.26)$$

Thus, the differential of f becomes:

$$df = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right) dz + \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) dz^*. \quad (2.27)$$

Finally, we only have to expand the differential of f but depending on the complex conjugate variables z and z^* ,

$$df(z, z^*) = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^*. \quad (2.28)$$

Identifying terms between (2.28) and (2.27), we obtain finally the desired expressions (2.22) and (2.23). ■

These two expressions relate the \mathbb{R} and \mathbb{R}^* derivatives with the derivatives with respect to the real and imaginary parts of the complex variables. This duality gives name to the $\mathbb{C}\mathbb{R}$ calculus.

Example 2.3 Assuming that z and z^* are independent, the following relations are straightforward:

$$\frac{\partial z}{\partial z} = \frac{1}{2} \left(\frac{\partial(x+jy)}{\partial x} - j \frac{\partial(x+jy)}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial x}{\partial x} + j \frac{\partial y}{\partial x} - j \frac{\partial x}{\partial y} + \frac{\partial y}{\partial y} \right) = 1, \quad (2.29)$$

$$\frac{\partial z^*}{\partial z^*} = \frac{1}{2} \left(\frac{\partial(x-jy)}{\partial x} + j \frac{\partial(x-jy)}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial x}{\partial x} - j \frac{\partial y}{\partial x} + j \frac{\partial x}{\partial y} + \frac{\partial y}{\partial y} \right) = 1, \quad (2.30)$$

$$\frac{\partial z}{\partial z^*} = \frac{1}{2} \left(\frac{\partial(x+jy)}{\partial x} + j \frac{\partial(x+jy)}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial x}{\partial x} + j \frac{\partial y}{\partial x} + j \frac{\partial x}{\partial y} - \frac{\partial y}{\partial y} \right) = 0, \quad (2.31)$$

$$\frac{\partial z^*}{\partial z} = \frac{1}{2} \left(\frac{\partial(x-jy)}{\partial x} - j \frac{\partial(x-jy)}{\partial y} \right) = \frac{1}{2} \left(\frac{\partial x}{\partial x} - j \frac{\partial y}{\partial x} - j \frac{\partial x}{\partial y} - \frac{\partial y}{\partial y} \right) = 0. \quad (2.32)$$

The Cauchy-Riemann condition revisited

Using the Wirtinger derivatives, we can reformulate the Cauchy-Riemann equations into a single condition. That illustrates the elegance of $\mathbb{C}\mathbb{R}$ Calculus.

Theorem 2.4 (The Cauchy-Riemann condition)

Let $f(z)$ be a scalar function of complex variable z . Then, f is holomorphic (complex analytic in z) if, and only if, it does not depend on the conjugate variable z^* . That is:

$$\frac{\partial f}{\partial z^*} = 0. \quad (2.33)$$

Proof: Simply applying Cauchy-Riemann equations (2.11) and (2.12) on condition (2.23), we can see that

$$\frac{\partial f}{\partial z^*} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) = \frac{1}{2} \left(\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) + j \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right) = 0. \quad (2.34)$$

One can see this from another point of view. It is well known (Kreyszig, 1988) that a holomorphic function always admits a power series expansion like

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n, \quad (2.35)$$

where $a_n, z_0 \in \mathbb{C}$. If any holomorphic function can be expressed in such way, it is clear that those functions does not depend on z^* .

Corollary of Theorem 2.4

The \mathbb{R} derivative of a function, $\partial f / \partial z$, is identical to the complex classical derivative $f'(z)$ as defined in (2.6), when $f(z, z^*)$ does not depends on z^* . This happens when f is holomorphic:

$$f(z) \text{ is holomorphic} \Leftrightarrow \frac{\partial f}{\partial z^*} = 0 \Leftrightarrow \frac{\partial f}{\partial z} = f'(z) \quad (2.36)$$

Proof: Substitute (2.33) into (2.28). ■

Rules of complex derivation

Here are some nice properties of Wirtinger derivatives.

$$\frac{\partial f^*}{\partial z^*} = \left(\frac{\partial f}{\partial z} \right)^*, \quad (2.37)$$

$$\frac{\partial f^*}{\partial z} = \left(\frac{\partial f}{\partial z^*} \right)^*, \quad (2.38)$$

$$df = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^* \quad (\text{Differential Rule}), \quad (2.39)$$

$$\frac{\partial h(g(z, z^*))}{\partial z} = \frac{\partial h}{\partial g} \frac{\partial g}{\partial z} + \frac{\partial h}{\partial g^*} \frac{\partial g^*}{\partial z} \quad (\text{Chain Rule}), \quad (2.40)$$

$$\frac{\partial h(g(z, z^*))}{\partial z^*} = \frac{\partial h}{\partial g} \frac{\partial g}{\partial z^*} + \frac{\partial h}{\partial g^*} \frac{\partial g^*}{\partial z^*} \quad (\text{Chain Rule}), \quad (2.41)$$

$$f(z) \in \mathbb{R} \Leftrightarrow \frac{\partial f}{\partial z^*} = \left(\frac{\partial f}{\partial z} \right)^*. \quad (2.42)$$

All of these properties extend naturally to the multivariate case, substituting the scalars by vectors and derivatives by gradients. The next subsection extends the

previous results to the multivariate case.

Multivariate $\mathbb{C}\mathbb{R}$ calculus

Before moving forward, some definitions from complex matrix algebra are needed.

Definition (Symmetric matrix): A *symmetric matrix* \mathbf{A} is a square matrix that satisfies

$$\mathbf{A}^T = \mathbf{A}, \quad (2.43)$$

where $(\cdot)^T$ denotes the transpose operator.

Definition (Hermitian matrix): A *Hermitian matrix* (or *self-adjoint matrix*) \mathbf{A} is a square matrix that satisfies

$$\mathbf{A}^H = \mathbf{A}, \quad (2.44)$$

where $(\cdot)^H$ denotes the Hermitian operator (complex conjugate transpose). They are the complex extension of a symmetric matrix. Symmetric and Hermitian matrices admit a simplified factorization that we will see later in this chapter.

Definition (Orthogonal matrix): An *orthogonal matrix* \mathbf{A} is a square matrix that satisfies

$$\mathbf{A}^T = \mathbf{A}^{-1} \Leftrightarrow \mathbf{A}^T \mathbf{A} = \mathbf{I}, \quad (2.45)$$

where \mathbf{I} is the *identity matrix*.

Definition (Unitary matrix): A *unitary matrix* \mathbf{A} is a square matrix that satisfies

$$\mathbf{A}^H = \mathbf{A}^{-1} \Leftrightarrow \mathbf{A}^H \mathbf{A} = \mathbf{I}. \quad (2.46)$$

Unitary matrices preserve *inner products*, i.e., $\langle \mathbf{c}, \mathbf{b} \rangle = \mathbf{c}^H \mathbf{b} = \langle \mathbf{A}\mathbf{c}, \mathbf{A}\mathbf{b} \rangle$. They also do not change the *norm*³ of a vector, $\|\mathbf{c}\| = \|\mathbf{A}\mathbf{b}\| = \|\mathbf{b}\|$. Orthogonal and unitary matrices are usually related to rotations and reflections in the vector space \mathbb{R}^n or \mathbb{C}^n , respectively.

We focus now on the multivariate case with real-valued scalar functions of complex multivariate variables. Indeed, the model for the cost function is:

$$f(\mathbf{z}): \mathbb{C}^n \rightarrow \mathbb{R}, \quad (2.47)$$

³ The norm of a vector \mathbf{g} is an scalar measure. The most known family is the set of p -norms, defined as $\|\mathbf{g}\|_p = \sqrt[p]{\sum_{i=1}^n |g_i|^p}$. If there is no sub-index, by default it will be the 2-norm (also called Euclidean norm). We will use them extensively in future chapters.

where

$$\mathbf{z} = \begin{bmatrix} z_i \\ \vdots \\ z_n \end{bmatrix} \in \mathbb{C}^n. \quad (2.48)$$

Definition (gradient and conjugate cogradient): Let f be a real-valued function whose independent variable is a complex vector \mathbf{z} , as defined in (2.47). Then, the *gradient* (\mathbb{R} gradient) and *conjugate cogradient* (\mathbb{R}^* gradient) are defined as in (Kreutz–Delgado, 2006):

$$\nabla_{\mathbf{z}} f(\mathbf{z}) = \left[\frac{\partial f}{\partial z_i} \right]_i \in \mathbb{C}^n, \quad (2.49)$$

$$\nabla_{\mathbf{z}^*} f(\mathbf{z}) = \left[\frac{\partial f}{\partial z_i^*} \right]_i \in \mathbb{C}^n. \quad (2.50)$$

The relations of these operators with respect to the gradients of the real and imaginary parts are extensions of the Wirtinger derivatives (2.22) and (2.23):

$$\nabla_{\mathbf{z}} = \frac{1}{2} (\nabla_{\mathbf{x}} - j \nabla_{\mathbf{y}}), \quad (2.51)$$

$$\nabla_{\mathbf{z}^*} = \frac{1}{2} (\nabla_{\mathbf{x}} + j \nabla_{\mathbf{y}}). \quad (2.52)$$

Example 2.4 The gradient and conjugate cogradient of the real-valued function $f(\mathbf{z}, \mathbf{z}^*) = \mathbf{z}^H \mathbf{z} = z_1 z_1^* + z_2 z_2^*$, are:

$$\nabla_{\mathbf{z}} f = \begin{bmatrix} z_1^* \\ z_2^* \end{bmatrix} = \mathbf{z}^*, \quad (2.53)$$

$$\nabla_{\mathbf{z}^*} f = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \mathbf{z}. \quad (2.54)$$

It is straightforward to see that Cauchy-Riemann equations, holomorphic conditions, rules of complex derivation, and the rest of the concepts presented in the previous sections, are naturally extended to the multivariate case. For example, the Cauchy-Riemann condition for f to be holomorphic is:

$$\nabla_{\mathbf{z}^*} f = \mathbf{0} \in \mathbb{C}^n. \quad (2.55)$$

Hereinafter, we will work with real-valued scalar cost functions of complex multivariate variables. These cost functions are defined so that they have extreme points that correspond to local solutions of a certain criterion.

2.4 Optimization with Complex Variables

In this section we start to apply the concepts defined above. Unless otherwise is stated, we will deal with scalar real-valued functions defined as:

$$f(\mathbf{z}): \mathbb{C}^n \rightarrow \mathbb{R}. \quad (2.56)$$

2.4.1 Stationary points

The function f can have an arbitrary shape, but as a function of a complex vector, it can be seen as a group of mountains in a multidimensional space. This landscape has valleys and peaks, corresponding to local maxima and minima of the real-valued function. These points are called *extreme points* or *stationary points*, and they share a nice property: the \mathbb{R} and \mathbb{R}^* gradients of the cost function vanish at them.

Theorem 2.5 (Extreme points)

Let f be a real-valued function as defined in (2.56). Then, the following two conditions are, each one, necessary and sufficient in order to optimize f with respect to a complex vector \mathbf{z} . At the extreme point $\mathbf{z} = \mathbf{z}_e$, it holds:

$$\nabla_{\mathbf{z}} f(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{z}_e} = \mathbf{0}, \quad (2.57)$$

$$\nabla_{\mathbf{z}^*} f(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{z}_e} = \mathbf{0}. \quad (2.58)$$

Proof: This result is just a multidimensional extension to the well-known result for scalar complex variables, where the extreme points of a function f defined as $f(z): \mathbb{C} \rightarrow \mathbb{R}$, are found when:

$$\frac{\partial f(z)}{\partial z} \Big|_{z=z_e} = 0, \quad (2.59)$$

$$\frac{\partial f(z)}{\partial z^*} \Big|_{z=z_e} = 0. \quad (2.60)$$

■

Any algorithm that optimizes the cost function f should reach one of these extreme points, where the criterion represented by the function is *maximized*, *minimized*, or reaches an *inflexion point*.

While we change the vector \mathbf{z} , we are changing the real value of the cost function f . For each \mathbf{z} , the value of f is determined. But, the reverse does not always hold, so f is not an injective function, in general.

This means that we can move freely the vector in any direction, as we were walking on the mountains, and watch for the effects on the objective function (that would be the height over the multidimensional surface).

An interesting question is: which direction is the one that achieves the maximum rate of change? If we are on the slopes of a valley, that direction leads us directly to the local minimum.

2.4.2 The direction of maximum rate of change

To answer that question, we look at how a small change on the vector variable is translated to the value of the cost function. The main result of this section is stated in the following theorem:

Theorem 2.6 (Direction of maximum rate of change)

Let $f(\mathbf{z}): \mathbb{C}^n \rightarrow \mathbb{R}$ be a real-valued function of complex multivariate variable. Then, the direction of maximum rate of change is given by

$$\nabla_{\mathbf{z}} f(\mathbf{z}). \quad (2.61)$$

And thus, moving \mathbf{z} in the same direction of (2.61), we are maximizing the cost function f . On the other hand, moving in the opposite direction $-\nabla_{\mathbf{z}} f(\mathbf{z})$, we are minimizing the cost function f .

Proof: Using the differential rule (2.39) for vectors, we obtain:

$$df = \left(\nabla_{\mathbf{z}} f(\mathbf{z})\right)^T d\mathbf{z} + \left(\nabla_{\mathbf{z}} f(\mathbf{z})\right)^T d\mathbf{z}^* \in \mathbb{R}. \quad (2.62)$$

Identifying the expression (2.4) of the real part of a complex variable:

$$df = 2\Re\left\{\left(\nabla_{\mathbf{z}} f(\mathbf{z})\right)^T d\mathbf{z}\right\}. \quad (2.63)$$

Using the multivariate equivalent of property (2.42) for real-valued functions, (2.63) becomes:

$$df = 2\Re \left\{ \left(\nabla_z f(\mathbf{z}) \right)^H dz \right\} = 2 \left\langle \nabla_z f(\mathbf{z}), dz \right\rangle. \quad (2.64)$$

This expression is proportional to the scalar product of two complex-valued vectors, $\nabla_z f(\mathbf{z})$ and dz . From basic geometry (Arfken, 1985), we know that the scalar product ($\langle \cdot, \cdot \rangle$) is maximized when the two vectors has the same direction, and minimized when they have opposite directions. In general, we are interested on minimizing the cost functions, because they usually represent an undesirable quality or an error of the system. ■

Another interesting situation occurs when vectors $\nabla_z f(\mathbf{z})$ and dz are orthogonal. The scalar product of two orthogonal vector is null. We can interpret this fact as if the rate of change df vanishes, so the cost function does not change.

It is interesting to see that, obviously, the *isobars* of the cost functions are defined by this situation. In fact, the locus of points orthogonal to the vector $\nabla_z f(\mathbf{z})$ locally define the points where the cost function f takes the same value.

Note that the same reasoning cannot be done for the gradient $\nabla_z f(\mathbf{z})$. Looking at (2.63), it is easy to see that it does not represent an scalar product, so it does not gives any insight about the geometry of the gradient of the cost function.

Example 2.5 Let $f : \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{R}$ be the squared Euclidean distance to the origin, given by

$$f(z, z^*) = |z|^2 = zz^*. \quad (2.65)$$

We can only display on the complex plane the level curves of a scalar real function of complex scalar variable. Thus, the resulting plot is shown in the Figure 2.1.

The formal \mathbb{R} and \mathbb{R}^* derivatives of this function are, assuming that z and z^* are independent,

$$\frac{\partial f}{\partial z} = z^*, \quad (2.66)$$

$$\frac{\partial f}{\partial z^*} = z. \quad (2.67)$$

These two derivatives are also shown in Figure 2.1. They are represented by an arrow marking the direction of change of both derivatives. In this case, note that they are mutually orthogonal, but this is not a general property.

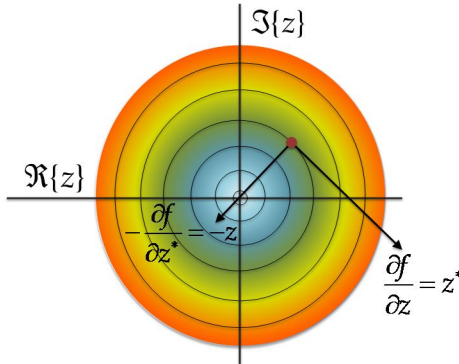


Figure 2.1 - This is a contour plot of the function defined in (2.65), and used in Example 2.5. Blue circumferences correspond to lower values of the cost function, where red circumferences correspond to high values.

It is easy to see that the direction of maximum rate of change corresponds to $\partial f / \partial z^*$, and its orthogonal is the direction of no change. By chance, the value of $\partial f / \partial z$ is this orthogonal direction, but that does not happen in most of the cases. The value of

$$-\frac{\partial f}{\partial z^*} \quad (2.68)$$

represents the direction of maximum descent, and the direction orthogonal to it represents the direction where the cost function f locally does not change its value.

2.4.3 Function minimization via a steepest descent algorithm

Imagine that we have the objective of minimizing the cost function $f(\mathbf{z}) : \mathbb{C}^n \rightarrow \mathbb{R}$. We must find the optimal vector \mathbf{z} that achieves the global minimum of f ,

$$\mathbf{z}_{opt} = \arg \left\{ \min_{\mathbf{z}} f(\mathbf{z}) \right\}. \quad (2.69)$$

Suppose now that we develop an iterative algorithm, where at each iteration we compute a new refined vector $\mathbf{z}^{(k+1)}$, using the previous version of the vector $\mathbf{z}^{(k)}$. Ideally, the function f evaluated at the new point should be smaller than the previous one, because that would indicate that we are minimizing the objective function,

$$f(\mathbf{z}^{(k+1)}) \leq f(\mathbf{z}^{(k)}). \quad (2.70)$$

To update the desired vector, we need additional information. That information can be the direction of *steepest descent* $\mathbf{d}^{(k)}$, so adding a scaled version of it would lead from $\mathbf{z}^{(k)}$ to $\mathbf{z}^{(k+1)}$, iteration by iteration,

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \mu^{(k)} \mathbf{d}^{(k)}. \quad (2.71)$$

The scale is controlled by the non-negative real parameter $\mu^{(k)} \in \mathbb{R}^+$, that receives the name of *step size*. It is responsible of the speed of convergence of the algorithm, and may cause it to fail if it is sufficiently large. On the other hand, if it is sufficiently small and the algorithm is not well initialized, it can cause to get stuck into a local minimum or a saddle point (provided that these exist in the cost function).

We will go deeper into this behavior in Chapter 5, but an intuitive discussion of this topic is shown in Figure 2.2. The left one shows a slow but direct path to the solution (the central point), that corresponds to a small step size. The right one shows a fast but chaotic convergence, related to a large step size. A balance between both extremes is usually the most reliable step size.

As seen in Theorem 2.6, the direction of the steepest descent is given by the opposite of the conjugate cogradient,

$$\mathbf{d}^{(k)} = -\nabla_{\mathbf{z}^*} f(\mathbf{z}^{(k)}). \quad (2.72)$$

So, we have that the update equation for a steepest descent algorithm is

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - \mu^{(k)} \nabla_{\mathbf{z}^*} f(\mathbf{z}^{(k)}). \quad (2.73)$$

Algorithms like this one have some advantages. On one hand, a closed-form solution for \mathbf{z}_{opt} may not exist, but if we can find the conjugate cogradient (2.61), we can run the steepest descent algorithm to reach the optimal point. On the other hand, those algorithms are also adaptive in *non-stationary* situations, where the statistical properties of the vectors involved changes over time.

The main drawback is that it is necessary to select carefully the step size $\mu^{(k)}$ to have good chances of success. Fortunately, there are both empirical and theoretical rules to select this parameter (see (Diniz, 1997)). In Chapter 5 we return to this interesting topic.

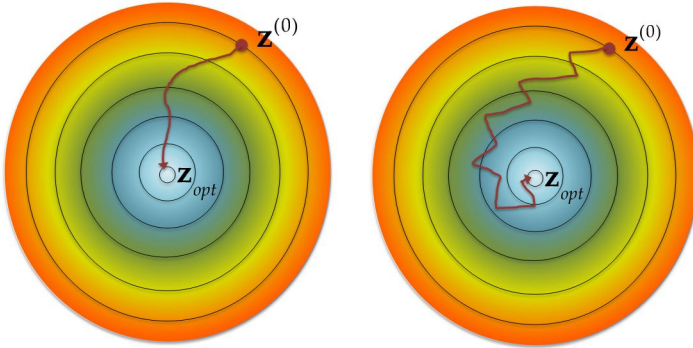


Figure 2.2 - This is a contour plot of the cost function $f(z, z^*) = |z|^2$, using two types of the step size $\mu^{(k)}$.

2.5 Augmented Complex Descriptions

In this section, we will give an overview of a richer approach to develop tools for complex-valued signal processing. This new point of view aims to exploit more information about the signal than the classical approaches. Most of this section is based upon the research compilation (Schreier, 2010).

2.5.1 Properness and circularity

All the signals involved here are *random processes*, that is, the value of each sample is a *random variable*. In the past, it has often been assumed implicitly that complex random signals are simpler than they really are. We start working with complex scalar random variables, but the extension to vectors is straightforward.

Definition (Proper complex random variable): Given a complex random variable $z \in \mathbb{C}$, we say that z is *proper* if it is uncorrelated with its complex conjugate, that is,

$$E\{z(z^*)^*\} = E\{zz\} = 0, \quad (2.74)$$

where $E\{\cdot\}$ is the statistical expectation operator. The concept of properness was first proposed in (Neeser, 1993). A deeper insight establishes that properness can be seen as a second-order implication of a more restrictive constraint.

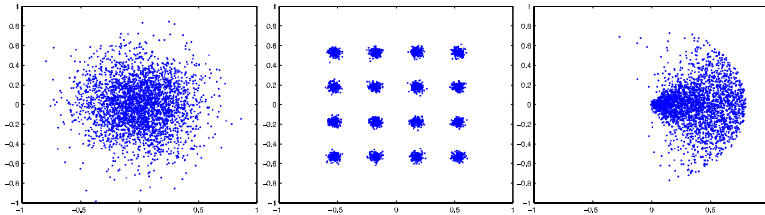


Figure 2.3 - Scatter plots for (left) circular, (center) proper but non-circular, and (right) improper, and thus non-circular, data.

There are situations where not only the second order cross-moment between z and z^* is zero, but the whole probability distribution is invariant under rotations in the complex plane. These complex random signals are said to be circular.

Definition (Circular complex random variable): Given a complex random variable $z \in \mathbb{C}$, with probability density function (p.d.f.), $p(z)$, we say that z is circular if $p(z)$ is invariant to rotations in \mathbb{C} , that is,

$$p(z) = p(ze^{j\theta}), \quad \forall \theta \in [0, 2\pi). \quad (2.75)$$

This latter definition imposes a very strong structure over the desired complex random variable. Indeed, that means that all the cross-moments between z and z^* are null, not only the one corresponding to the second order.

Note that the following implication holds, but the contrary is not always true:

$$z \text{ circular} \Rightarrow z \text{ proper}. \quad (2.76)$$

For this reason, properness is also called *second order circularity*. In Figure 2.3, three examples of complex random variables are shown. The first one (left) can be drawn from a circular Gaussian complex random variable, and it is a typical scatter plot of a radar application. It can be seen that the probability density function is invariant to rotations on the complex plane. The variable is, in addition, proper.

The second scatter plot (center) shows a *Quadrature Phase-Shift Keying (QPSK)* signal imbued in *Additive White Gaussian Noise (AWGN)*, which is the typical signal found in the input of a *decisor* in a receiver chain. Its p.d.f. is clearly dependent on rotations, so the variable is non-circular. However, if we compute the correlation (2.74), it is zero, so the variable is still proper.

The third one (right) is a typical scatter plot than can be found in advanced

signal processing applications from real world. This time, the correlation between the complex variable and its conjugate, is not zero. It is an improper, and thus non-circular, complex random variable. This latter statement can be corroborated by inspecting the scatter plot: the p.d.f. is clearly dependent on rotations.

Until now, we have not worried about the properness or circularity of the signals that we use, but this information would reveal to be very useful when designing signal processing algorithms.

2.5.2 Connection between real and complex descriptions

Let Ω be the sample space of a random experiment, and let $\mathbf{z} : \Omega \rightarrow \mathbb{C}^n$ be the *complex* random vector defined as

$$\mathbf{z} = \mathbf{x} + jy, \quad (2.77)$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ are the real and imaginary parts of \mathbf{z} , respectively. From this expression, we define two additional representations of the same vector \mathbf{z} .

Definition (Real composite random vector): If we stack the real part over the imaginary part, we get the *real composite* random vector $\mathbf{z}_c : \Omega \rightarrow \mathbb{R}^{2n}$, defined as

$$\mathbf{z}_c = \begin{bmatrix} \Re\{\mathbf{z}\} \\ \Im\{\mathbf{z}\} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \quad (2.78)$$

This expression translates the complex random vector to the real domain, maintaining all the information but doubling the dimension of the array. The second additional representation is even more interesting.

Definition (Augmented complex random vector): If we stack the complex vector \mathbf{z} over the complex conjugate vector \mathbf{z}^* , we obtain the *augmented complex* random vector $\underline{\mathbf{z}} : \Omega \rightarrow \mathbb{C}_*^{2n}$,

$$\underline{\mathbf{z}} = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix}. \quad (2.79)$$

The space of all complex augmented vectors is denoted by \mathbb{C}_*^{2n} to distinguish from \mathbb{C}^{2n} , where the first and the second half of the vector are not related.

Note that augmented vectors have redundant information, but their structure reveals to be very elegant and useful to work with.

The augmented complex random vector $\underline{\mathbf{z}}$ is related to the real composite vector \mathbf{z}_R as

$$\underline{\mathbf{z}} = \mathbf{T}\mathbf{z}_C \Leftrightarrow \mathbf{z}_C = \frac{1}{2}\mathbf{T}^H\underline{\mathbf{z}}, \quad (2.80)$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix} \in \mathbb{C}^{2n \times 2n} \quad (2.81)$$

is a unitary (up to a factor of 2) *real-to-complex* transformation,

$$\mathbf{T}\mathbf{T}^H = \mathbf{T}^H\mathbf{T} = 2\mathbf{I}. \quad (2.82)$$

Widely linear transformations

In this section, we aim to introduce *widely linear transformations*, which are linear transformations in the complex domain that has a specific structure. It is well known that a complex matrix $\mathbf{H} \in \mathbb{C}^{m \times n}$ can define a *complex linear transformation* like

$$\mathbf{f} = \mathbf{u} + j\mathbf{v} = \mathbf{H}\mathbf{z} \in \mathbb{C}^m. \quad (2.83)$$

If a *real composite linear* transformation $\mathbf{M} \in \mathbb{R}^{2m \times 2n}$ is applied to the composite real vector \mathbf{z}_C , we obtain another composite real vector $\mathbf{f}_C : \Omega \rightarrow \mathbb{R}^{2m}$,

$$\mathbf{f}_C = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{M}\mathbf{z}_C, \quad (2.84)$$

where the blocks are $\mathbf{M}_{ij} \in \mathbb{R}^{m \times n}$.

The augmented complex version of \mathbf{f}_C is

$$\underline{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} = \mathbf{T}\mathbf{f}_C = \mathbf{T} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{T}\mathbf{M}\mathbf{z}_C. \quad (2.85)$$

And now, by using (2.82) and identifying,

$$\underline{\mathbf{f}} = \left(\frac{1}{2}\mathbf{T}\mathbf{M}\mathbf{T}^H \right) (\mathbf{T}\mathbf{z}_C) = \underline{\mathbf{H}}\underline{\mathbf{z}}. \quad (2.86)$$

The matrix $\underline{\mathbf{H}}$ relates linearly two augmented complex representations. It has important properties that must be highlighted.

Definition (Augmented matrix): A matrix $\underline{\mathbf{H}}$ is an *augmented matrix* if it satisfies a particular block pattern,

$$\underline{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_2^* & \mathbf{H}_1^* \end{bmatrix}, \quad (2.87)$$

where the blocks are related to the blocks \mathbf{M}_{ij} as

$$\mathbf{H}_1 = \frac{1}{2} [\mathbf{M}_{11} + \mathbf{M}_{22} + j(\mathbf{M}_{21} - \mathbf{M}_{12})], \quad (2.88)$$

and

$$\mathbf{H}_2 = \frac{1}{2} [\mathbf{M}_{11} - \mathbf{M}_{22} + j(\mathbf{M}_{21} + \mathbf{M}_{12})]. \quad (2.89)$$

The matrix

$$\underline{\mathbf{H}} = \left(\frac{1}{2} \mathbf{TMT}^H \right) \in \mathbb{C}_*^{2m \times 2n} \quad (2.90)$$

is the augmented description of a *widely linear* (or *linear-conjugate-linear*) transformation

$$\mathbf{f} = \mathbf{H}_1 \mathbf{z} + \mathbf{H}_2 \mathbf{z}^*, \quad (2.91)$$

that degenerates to a complex linear transformation $\mathbf{f} = \mathbf{H}_1 \mathbf{z}$ when $\mathbf{H}_2 = \mathbf{0}$. This imposes a particular structure on the real matrix \mathbf{M} .

A complex linear transformation $\mathbf{f} = \mathbf{H}_1 \mathbf{z}$ (not widely linear) has a real composite linear representation like

$$\mathbf{f}_C = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ -\mathbf{M}_{12} & \mathbf{M}_{11} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \mathbf{Mz}_C. \quad (2.92)$$

This relation is the result of forcing (2.89) to be zero.

Note that complex linear transformations on \mathbb{R}^{2n} , like (2.84), are linear only on \mathbb{C}^n if they satisfy a certain structure, stated in (2.92). Otherwise, the equivalent operation on \mathbb{C}^n is widely linear. The Figure 2.4 summarizes the discussion exposed in this section.

2.5.3 Second-order augmented statistics

Let us consider the real composite random vector \mathbf{z}_C . Its *mean* vector is

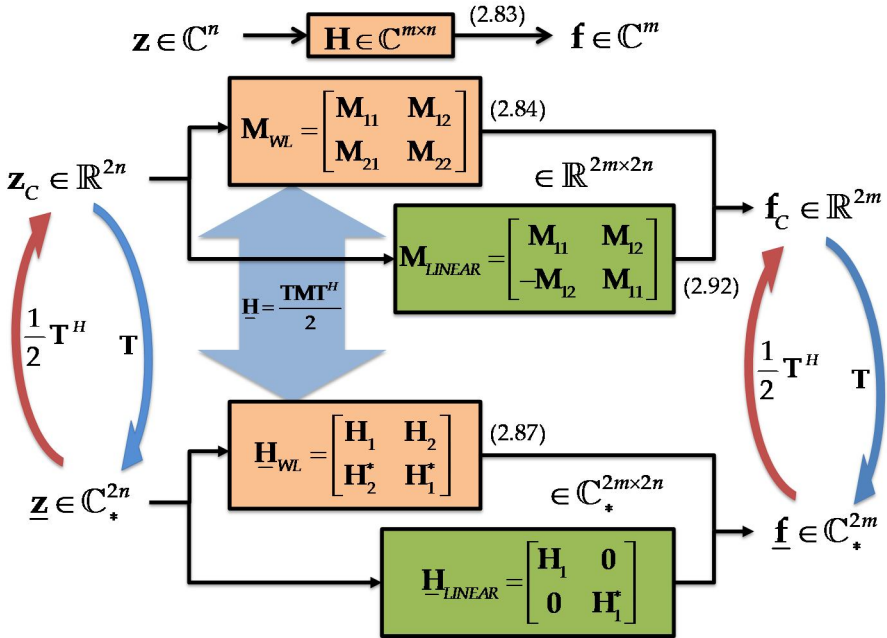


Figure 2.4 – This block diagram summarizes the discussion presented in Section 2.5.2. Also, we provide the relations between different representations and the equations where they are defined.

$$\mu_{z_c} = E\{\mathbf{z}_c\} = \begin{bmatrix} \mu_u \\ \mu_v \end{bmatrix} \in \mathbb{R}^{2m \times 2n} \quad (2.93)$$

and its *covariance* matrix is

$$\mathbf{R}_{z_c} = E\left\{\left(\mathbf{z}_c - \mu_{z_c}\right)\left(\mathbf{z}_c - \mu_{z_c}\right)^T\right\} = \begin{bmatrix} \mathbf{R}_u & \mathbf{R}_{uv} \\ \mathbf{R}_{uv}^T & \mathbf{R}_v \end{bmatrix}, \quad (2.94)$$

where \mathbf{R}_u , \mathbf{R}_v and \mathbf{R}_{uv} are the covariance and *cross-covariance* matrices of their respective real random vectors. Let us look at the augmented representation. The *augmented mean vector* is

$$\underline{\mu}_z = E\{\underline{\mathbf{z}}\} = \mathbf{T}\mathbf{z} = \begin{bmatrix} \mu_z \\ \mu_z^* \end{bmatrix} \in \mathbb{C}_*^{2m \times 2n}, \quad (2.95)$$

and the *augmented covariance matrix* is

$$\underline{\mathbf{R}}_z = E \left\{ \left(\underline{\mathbf{z}} - \underline{\mu}_z \right) \left(\underline{\mathbf{z}} - \underline{\mu}_z \right)^H \right\} = E \left\{ \begin{bmatrix} \mathbf{z} - \mu_z \\ \mathbf{z}^* - \mu_z^* \end{bmatrix} \begin{bmatrix} \mathbf{z}^* - \mu_z^* & , & \mathbf{z} - \mu_z \end{bmatrix} \right\}. \quad (2.96)$$

Note that this augmented covariance matrix results to be Hermitian

$$\underline{\mathbf{R}}_z = \begin{bmatrix} \mathbf{R}_z & \tilde{\mathbf{R}}_z \\ \tilde{\mathbf{R}}_z^* & \mathbf{R}_z^* \end{bmatrix} = \underline{\mathbf{R}}_z^H. \quad (2.97)$$

The upper left block of (2.97) is the usual covariance matrix

$$\mathbf{R}_z = E \left\{ \left(\mathbf{z} - \mu_z \right) \left(\mathbf{z} - \mu_z \right)^H \right\} = \mathbf{R}_z^H, \quad (2.98)$$

and the lower right block is the symmetric *pseudo-covariance* matrix (also known as the *complementary covariance matrix*, or the *conjugate covariance matrix*),

$$\tilde{\mathbf{R}}_z = E \left\{ \left(\mathbf{z} - \mu_z \right) \left(\mathbf{z} - \mu_z \right)^T \right\} = \tilde{\mathbf{R}}_z^T \in \mathbb{C}^{n \times n}. \quad (2.99)$$

Note that the covariance matrix has real elements in its diagonal that can be interpreted as the *power* of the components, but this is not true in general for the pseudo-covariance matrix.

Hereinafter, we will suppose that the random vectors have zero mean. As we see in (2.97), both the covariance and the pseudo-covariance matrices are required for a complete description of the second-order statistics of \mathbf{z} .

This is not the usual practice in classical complex-valued statistics, because for proper processes, only the covariance matrix is necessary to describe the whole second-order statistics. Thus, for non-proper processes we are missing valuable information if we ignore the pseudo-covariance matrix, as we will see further in this section.

The special case where the structure of \mathbf{z} itself forces the pseudo-covariance matrix to be null is summarized in the following theorem.

Theorem 2.7 (Properness condition)

Let $\mathbf{z} = \mathbf{x} + jy \in \mathbb{C}^n$ be a complex random vector. Then, if its pseudo-covariance matrix vanishes, \mathbf{z} is proper. Otherwise, \mathbf{z} is called *improper* as we saw in Section 2.5.1, and

$$\tilde{\mathbf{R}}_z = \mathbf{0}. \quad (2.100)$$

This happens when the covariance matrix of the real and imaginary parts are, respectively,

$$\mathbf{R}_x = \mathbf{R}_y, \quad (2.101)$$

$$\mathbf{R}_{xy} = -\mathbf{R}_{xy}^T. \quad (2.102)$$

The second-order statistics are fully described by the covariance matrix,

$$\mathbf{R}_z = 2\mathbf{R}_x - 2j\mathbf{R}_{xy} = 2\mathbf{R}_y + 2j\mathbf{R}_{xy}^T, \quad (2.103)$$

so the augmented covariance $\underline{\mathbf{R}}_z$ matrix is block-diagonal.

Proof: Substituting the definition (2.77) in (2.99),

$$\begin{aligned} \tilde{\mathbf{R}}_z &= E \left\{ \left((\mathbf{x} - \mu_x) + j(\mathbf{y} - \mu_y) \right) \left((\mathbf{x} - \mu_x) + j(\mathbf{y} - \mu_y) \right)^T \right\} \\ &= E \left\{ (\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T \right\} - E \left\{ (\mathbf{y} - \mu_y)(\mathbf{y} - \mu_y)^T \right\} \\ &\quad + jE \left\{ (\mathbf{y} - \mu_y)(\mathbf{x} - \mu_x)^T \right\} + jE \left\{ (\mathbf{x} - \mu_x)(\mathbf{y} - \mu_y)^T \right\} \\ &= \mathbf{R}_x - \mathbf{R}_y + j(\mathbf{R}_{xy}^T + \mathbf{R}_{xy}) = \tilde{\mathbf{R}}_z^T. \end{aligned} \quad (2.104)$$

And looking at the structure, one can check that $\tilde{\mathbf{R}}_z = \mathbf{0}$ if, and only if, $\mathbf{R}_x = \mathbf{R}_y$ and $\mathbf{R}_{xy} = -\mathbf{R}_{xy}^T$. These are the conditions (2.101) and (2.102).

Simultaneously, we can expand the covariance matrix using the definition (2.98),

$$\begin{aligned} \mathbf{R}_z &= E \left\{ \left((\mathbf{x} - \mu_x) + j(\mathbf{y} - \mu_y) \right) \left((\mathbf{x} - \mu_x) + j(\mathbf{y} - \mu_y) \right)^H \right\} \\ &= E \left\{ (\mathbf{x} - \mu_x)(\mathbf{x} - \mu_x)^T \right\} + E \left\{ (\mathbf{y} - \mu_y)(\mathbf{y} - \mu_y)^T \right\} \\ &\quad + jE \left\{ (\mathbf{y} - \mu_y)(\mathbf{x} - \mu_x)^T \right\} - jE \left\{ (\mathbf{x} - \mu_x)(\mathbf{y} - \mu_y)^T \right\} \\ &= \mathbf{R}_x + \mathbf{R}_y + j(\mathbf{R}_{xy}^T - \mathbf{R}_{xy}) = \mathbf{R}_z^H. \end{aligned} \quad (2.105)$$

Substituting conditions for properness (2.101) and (2.102), we obtain the expression (2.103). When this happens, the augmented covariance matrix

(2.97) is block diagonal,

$$\underline{\mathbf{R}}_z = \begin{bmatrix} \mathbf{R}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_z^* \end{bmatrix}, \quad (2.106)$$

and has redundant information.

Note that \mathbf{x} and \mathbf{y} are real-valued random vectors, so the Hermitian and the transpose operators in definitions of \mathbf{R}_x and \mathbf{R}_y are interchangeable.

■

We address now a key problem in complex-valued signal processing. How to extend definitions from the real to the complex case?

To aim for the robustness of the theory, concepts and definitions should be equivalent for real and complex domains. Consider the definition of uncorrelatedness for real-valued random vectors,

$$\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n \text{ uncorrelated} \Leftrightarrow \mathbf{R}_{z_1 z_2} = \mathbf{0} \quad (2.107)$$

Note that, if \mathbf{z}_1 and \mathbf{z}_2 have zero mean, we can read (2.107) as orthogonality $\mathbf{z}_1 \perp \mathbf{z}_2$ in the Hilbert space of second-order real random variables.

Using the equivalency principle stated above, we cannot avoid the value of the pseudo-covariance. It must be zero too, to force *strong uncorrelatedness* in the complex domain. So, the equivalent condition of (2.107) for complex-valued random vectors is not only that the covariance matrix vanishes, but that the augmented covariance matrix does too,

$$\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{C}^n \text{ uncorrelated} \Leftrightarrow \underline{\mathbf{R}}_{z_1 z_2} = \mathbf{0}. \quad (2.108)$$

If \mathbf{z}_1 and \mathbf{z}_2 have zero mean, there is double orthogonality in the Hilbert space of second-order complex random variables, that is, $\mathbf{z}_1 \perp \mathbf{z}_2$ and $\mathbf{z}_1 \perp \mathbf{z}_2^*$ simultaneously.

2.6 Conclusions

In this chapter, we have reviewed the main mathematical tools for working in the complex domain. As we have seen, there are some behaviors and structures in the complex domain that do not follow the natural extension from the field of real numbers. There have been some key points that we would like to highlight here:

- We started by reviewing the properties of complex-valued functions

and variables. The rules for derivations with complex variable are not as straightforward as in the real case.

- Functions of interest in engineering are usually real-valued functions called cost functions, that are not holomorphic and must be treated by using the $\mathbb{C}\mathbb{R}$ - Calculus (or Wirtinger Calculus). The complex variable and its complex conjugate counterpart can be treated as independent variables, making easy to deal with them.
- In the surface of a cost function, the direction of maximum rate of change is given by the conjugate cogradient of the cost function, instead of the gradient. The locus where the function does not changes its value is locally orthogonal to this conjugate cogradient.
- Recently, the study of complex-valued signal processing has explored another point of view called augmented statistics. The main idea is that the second order approaches are lacking important information, and must be completed by taking into account possible correlations between conjugate parts. The new approach leads to the so-called widely linear models.

Most of these ideas have started to receive the interest of the signal processing community in the past decade. The results reviewed in this chapter are the base to active lines of research, like the one proposed in this Thesis. We use several of the previous concepts on the development of the tools presented in next chapters.

3 COMPLEX-VALUED SIGNAL PROCESSING

*Production is not the application of tools to materials, but
logic to work.*

- Peter F. Drucker -

Complex variables, vectors and random processes arise naturally to solve high challenging topics in different fields of the science like signal processing, communications and biomedical engineering. Some of these selected applications are extensively treated in the textbooks (Adali, 2010), (Mandic, 2009), (Hyvärinen, 2001) and (Schreier, 2010), which are also the basis for this chapter.

3.1 Introduction

In this chapter, we review both classical and novel applications in the field of signal processing when dealing with complex-valued signals. Once we have presented the fundamentals of complex variables, holomorphic functions, and complex derivatives, now we show how these tools can help when analyzing or designing signal processing systems.

The chapter is organized as follows. In Section 3.2, we review the complex version of a well-known supervised criterion that we reference along the entire Thesis. The adaptive algorithm to solve this criterion is provided in Section 3.3. Section 3.4 review the main ideas behind signal separation and extraction, and presents successful techniques to solve these problems. In Section 3.5 and Section 3.6, we show what happens when we apply the new tools from widely linear models and augmented statistics to these problems. Finally, Section 3.7 summarizes the conclusions.

3.2 The Minimum Mean Square Error Criterion

We introduce the complex version of the *Minimum Mean Square Error (MMSE)* criterion in the context of the linear estimation of a signal by passing another given signal through a filter. This criterion aims to minimize the estimation mismatch by means of the penalization of the errors in a quadratic form.

In *supervised* applications like this one, there is a small amount of information available about the desired signals, whose use will be only to *train* the system at an initial stage.

Suppose that a *linear filter* approximates (estimates) a *desired sequence* $d(k)$ through a linear combination of the *input sequence* $x(k)$. The estimate of the desired sequence is the *output sequence*

$$y(k) = \sum_{j=1}^m w_j^* x(k-j+1) = \mathbf{w}^H \mathbf{x}(k), \quad (3.1)$$

where

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-m+1) \end{bmatrix} \in \mathbb{C}^m \quad (3.2)$$

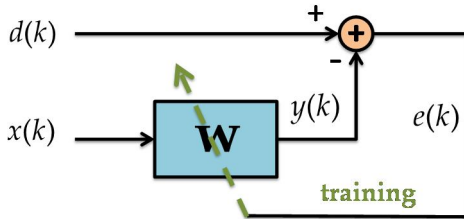


Figure 3.1 - Block system of the sequence estimation problem. In the figure, $d(k)$ is the desired sequence that we want to estimate. $x(k)$ is the input sequence that we will use, by passing it by a linear system \mathbf{w} , resulting in the estimated sequence $y(k)$. The error between the estimate and the desired sequences is the estimation error $e(k)$, and it is used to design the coefficients of the filter.

is a vector whose elements are delayed copies of the input sequence, and

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{m-1} \end{bmatrix} \in \mathbb{C}^m \quad (3.3)$$

is the linear filter. The Figure 3.1 shows a block diagram of this basic system.

The input sequence $x(k)$ is intended to be *Wide Sense Stationary (WSS)*, which intuitively means that its first and second order statistical properties do not change over time (Priestley, 1981).

We also have an *estimation error* $e(k)$ that is the difference between the desired and the output sequence:

$$e(k) = d(k) - y(k) \in \mathbb{C}. \quad (3.4)$$

To achieve an accurate estimation, we need to minimize the estimation error. A common criterion, widely used due to its easy mathematical treatment and its intuitive interpretation, is to minimize the *mean square* of the estimation error.

Definition (The Mean Square Error (MSE)): given a complex error sequence $e(k)$, the cost function associated to the *Mean Square Error (MSE)* is

$$J_{MSE} = E \left\{ |e(k)|^2 \right\} = E \left\{ e(k) e^*(k) \right\} \in \mathbb{R}. \quad (3.5)$$

Note that, as we stated in Section 2.3, the cost function is a real-valued function of complex variable. Thus, it can be maximized or minimized, depending on our interest. As we have just said, in this case we are interested on minimizing the MSE.

Fortunately, we have already seen this kind of quadratic functions previously in Example 2.1 and Example 2.5, when we presented the fundamentals of complex derivatives.

Definition (The Minimum Mean Square Error (MMSE) criterion): By minimizing the MSE cost function (3.5) with respect to \mathbf{w} , we obtain the solution to the Minimum Mean Square Error (MMSE) criterion:

$$\mathbf{w}_{MMSE} = \arg \left\{ \min_{\mathbf{w}} J_{MSE}(\mathbf{w}) \right\}. \quad (3.6)$$

The solution of this unconstrained optimization problem is a well-known result on the signal processing literature, and it is presented in the following theorem.

Theorem 3.1 (Complex Wiener-Hopf solution of the MMSE criterion)

Given a linear filter \mathbf{w} that estimates a desired sequence $d(k)$ using a linear combination of input samples $x(k)$, the solution to the MMSE criterion is:

$$\mathbf{w}_{MMSE} = \mathbf{R}_x^{-1} \mathbf{r}_{xd}, \quad (3.7)$$

where

$$\mathbf{R}_x = E \left\{ \mathbf{x}(k) \mathbf{x}^H(k) \right\} \in \mathbb{C}^{m \times m} \quad (3.8)$$

is the non-singular input covariance matrix, and

$$\mathbf{r}_{xd} = E \left\{ d^*(k) \mathbf{x}(k) \right\} \in \mathbb{C}^m \quad (3.9)$$

is the cross-covariance between the input and the desired sequence. The input $\mathbf{x}(k)$ is defined as in (3.2). Note that the diagonal elements of \mathbf{R}_x are real, but this does not need to be true for the off-diagonal elements.

Proof: We take the gradient with respect to the conjugate vector \mathbf{w}^* , but keeping \mathbf{w} constant, just as we stated in the previous chapter. Indeed, at the optimal point $\mathbf{w} = \mathbf{w}_{MMSE}$, the condition (2.58) is fulfilled:

$$\left. \nabla_{\mathbf{w}^*} J_{MSE}(\mathbf{w}) \right|_{\mathbf{w}=\mathbf{w}_{MMSE}} = \mathbf{0}. \quad (3.10)$$

By computing the gradient with (3.4):

$$\begin{aligned}\nabla_{\mathbf{w}} \cdot J_{MSE}(\mathbf{w}) &= \nabla_{\mathbf{w}} \cdot \left(E \{ e(k) e^*(k) \} \right) = E \left\{ \nabla_{\mathbf{w}} \cdot \left(e(k) e^*(k) \right) \right\} \\ &= E \left\{ \nabla_{\mathbf{w}} \cdot \left((d(k) - y(k))(d^*(k) - y^*(k)) \right) \right\}.\end{aligned}\quad (3.11)$$

By using the definition of $y(t)$ from (3.1),

$$\nabla_{\mathbf{w}} \cdot J_{MSE}(\mathbf{w}) = E \left\{ \nabla_{\mathbf{w}} \cdot \left((d(k) - \mathbf{w}^H \mathbf{x}(k))(d^*(k) - \mathbf{w}^T \mathbf{x}^*(k)) \right) \right\}.\quad (3.12)$$

By expanding the product,

$$\begin{aligned}\nabla_{\mathbf{w}} \cdot J_{MSE}(\mathbf{w}) &= E \left\{ \nabla_{\mathbf{w}} \cdot (d(k)d^*(k)) \right\} - E \left\{ \nabla_{\mathbf{w}} \cdot (d(k)\mathbf{w}^T \mathbf{x}^*(k)) \right\} \\ &\quad + E \left\{ \nabla_{\mathbf{w}} \cdot (\mathbf{w}^H \mathbf{x}(k)\mathbf{w}^T \mathbf{x}^*(k)) \right\} - E \left\{ \nabla_{\mathbf{w}} \cdot (d^*(k)\mathbf{w}^H \mathbf{x}(k)) \right\}.\end{aligned}\quad (3.13)$$

The gradients of all the terms which does not depend on \mathbf{w}^* , vanish:

$$\nabla_{\mathbf{w}} \cdot J_{MSE}(\mathbf{w}) = E \left\{ \mathbf{x}(k)\mathbf{w}^T \mathbf{x}^*(k) \right\} - E \left\{ (d^*(k)\mathbf{x}(k)) \right\}\quad (3.14)$$

As we said in Section 2.4.1, this conjugate cogradient vanishes at the optimal point $\mathbf{w} = \mathbf{w}_{MMSE}$, so we only have to solve $\nabla_{\mathbf{w}} \cdot J_{MSE}(\mathbf{w}) = \mathbf{0}$ to obtain a well-known equation:

$$E \left\{ \mathbf{x}(k)\mathbf{x}^H(k) \right\} \mathbf{w}_{MMSE} = E \left\{ (d^*(k)\mathbf{x}(k)) \right\}.\quad (3.15)$$

Solving for \mathbf{w}_{MMSE} , we get (3.7). ■

To know how large is the value of the minimum of the cost function J_{\min} , we have to substitute the optimal filter \mathbf{w}_{MMSE} in (3.5) and expand the expression,

$$J_{\min} = J_{MSE}(\mathbf{w}_{MMSE}) = \sigma_d^2 - \mathbf{r}_{xd}^H \mathbf{R}_x^{-1} \mathbf{r}_{xd},\quad (3.16)$$

where $\sigma_d^2 = E \{ d(k)d^*(k) \}$ is the power of the desired sequence.

In addition, the cost function (3.5) can be rewritten in *quadratic form*,

$$J_{MSE}(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_{MMSE})^H \mathbf{R}_x (\mathbf{w} - \mathbf{w}_{MMSE})\quad (3.17)$$

an hence, it has a global minimum. See (Widrow, 1985) for a detailed derivation of these two expressions.

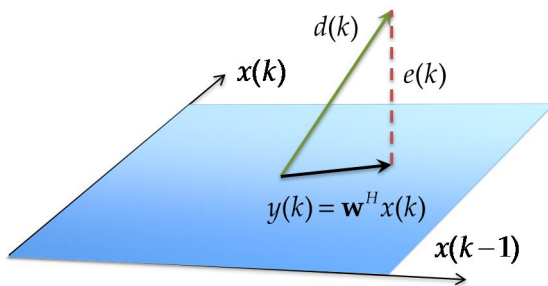


Figure 3.2 - This figure shows a two dimensional plane that represents the $\mathbf{x}(k)$ subspace. The solid line lying on the plane represents $y(k)$, the best possible approximation of the desired sequence $d(k)$ as a linear combination of $\mathbf{x}(k)$ with the weights \mathbf{w} . The difference between the desired and the output sequences is the error $e(k)$, which is orthogonal to the plane $\mathbf{x}(k)$ at the optimal solution.

3.2.1 Geometrical interpretation of the MMSE criterion

We know that the condition for the optimal point is that the conjugate cogradient of the cost function vanishes. If we cancel (3.14) and reorder,

$$E\{\mathbf{x}(k)(\mathbf{w}^T \mathbf{x}^*(k) - d^*(k))\} = -E\{\mathbf{x}(k)e^*(k)\} = 0, \quad (3.18)$$

leading to the so-called *orthogonality condition*,

$$E\{e^*(k)\mathbf{x}(k)\} = 0. \quad (3.19)$$

This condition has a nice physical interpretation: the output error of the filter and the input sequence subspace are orthogonal in the Hilbert space when the filter has converged to the optimal solution. When this happens, the estimated sequence $y(k)$ is the nearest possible (in Euclidean distance) to the desired sequence, laying in the input subspace $\mathbf{x}(k)$.

The error sequence is the difference of both sequences $y(k)$ and $d(k)$, and it is orthogonal to the input signal subspace. The Figure 3.2 resumes the previous discussion.

This optimization problem has a *closed-form solution*: an expression that gives us the optimal filter by direct computation. Note that, in practice, inversion of the covariance (3.8) can be very computationally expensive. Nevertheless, if the

sequences change their statistical properties over time, they are non-stationary and the solution (3.7) is suboptimal.

Suppose now that we are processing in real time the estimation $y(k)$ from the desired sequence $d(k)$ and the input $x(k)$. As we have seen, their properties are changing over time, so it would be a great idea if we synthesize the filter \mathbf{w} in an adaptive manner. Even the invertibility of \mathbf{R}_x is not guaranteed in this situation.

In the next section, we use instantaneous information of the sequences and develop an adaptive algorithm.

3.3 The Complex Least Mean Squares Algorithm

We continue studying the optimization of the MMSE criterion, but now in an adaptive and more efficient way. As we have seen in Section 2.4.3, the way to do that is to compute the conjugate cogradient and move towards its opposite direction to update the vector \mathbf{w} in the next iteration,

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mu^{(k)} \nabla_{\mathbf{w}^*} J_{LMS}(\mathbf{w}^{(k)}). \quad (3.20)$$

Note that, in this case, the cost function J_{LMS} is not the same as the MSE described in (3.5). The cost function for the *Complex Least Mean Squares (CLMS)* algorithm just drops the statistical expectation operator $E\{\cdot\}$ from the cost function. This means that, to advance to the next iteration $k+1$, we will only need information from the current iteration k . The cost function results:

$$J_{LMS}(\mathbf{w}) = |e(k)|^2 = |d(k) - \mathbf{w}^H \mathbf{x}(k)|^2. \quad (3.21)$$

This is the reason why this kind of algorithm is sometimes called *stochastic gradient descent algorithm*. The cost function (3.21) is a random variable and the result after each iteration is somehow unpredictable. It can be viewed as a rough estimation of the deterministic quantity (3.5) by a single instantaneous sample.

We just need now to compute the conjugate cogradient to obtain the update equation,

$$\nabla_{\mathbf{w}^*} J_{LMS}(\mathbf{w}) = \nabla_{\mathbf{w}^*} (e(k)e^*(k)) = e^*(k) \nabla_{\mathbf{w}^*} (e(k)). \quad (3.22)$$

And that is because \mathbf{w}^* and \mathbf{w} are supposed to be independent. As the conjugate cogradient of $e(k)$ is just $-\mathbf{x}(k)$, the equation (3.22) results

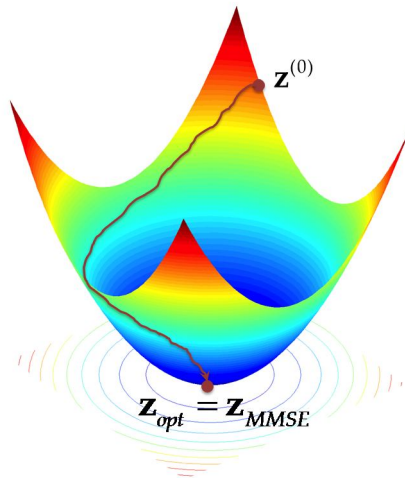


Figure 3.3 - This is a 3D plot of the cost function $J_{LMS}(\mathbf{w})$, versus the plane conformed by the coefficients of the filter \mathbf{w} .

$$\nabla_{\mathbf{w}} J_{LMS}(\mathbf{w}) = -e^*(k)\mathbf{x}(k). \quad (3.23)$$

By substituting in the update equation (3.20), we obtain the rule for the filter update of the CLMS algorithm,

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \mu^{(k)} e^*(k)\mathbf{x}(k). \quad (3.24)$$

This expression was first derived in (Widrow, 1975), and it has been widely used in practical applications, with many variants like *Normalized Least Mean Squares (NLS)* or *Recursive Least Squares (RLS)* that improve its performance or reliability (see (Hayes, 1996 p. 541) or (Górriz, 2010)).

3.3.1 Convergence of the CLMS algorithm

It is important to analyze the convergence of the algorithms. As the only parameter that we can control is the step size $\mu^{(k)}$, usually the analysis is completed with a condition for the convergence. If it is very small, the convergence could be slow. If it is greater than a certain condition, it may move chaotically around the solution, without reaching it.

An intuitive plot of the cost function $J_{LMS}(\mathbf{w})$ is shown in Figure 3.3. The lowest point of the basin is where the cost function achieves the optimal point for the filter coefficients. As we have seen, it matches the solution for the MMSE criterion.

It can be seen why this type of algorithms is called stochastic: the value of the function fluctuates, as it goes down, and eventually it stays fluctuating close the optimal point. Hopefully, the chaotic movement can be reduced by selecting the proper value for the step size $\mu^{(k)}$.

At the optimal point, the gradient (3.23) vanishes, i.e.,

$$e^*(k)\mathbf{x}(k) = 0, \quad (3.25)$$

and the filter approximates to its fixed point, called the Wiener-Hopf solution (Mandic, 2007):

$$\lim_{x \rightarrow \infty} (\mathbf{w}^{(k)}) = \mathbf{w}_{MMSE}. \quad (3.26)$$

The condition (3.25) can be averaged over all the realizations of the random processes, leading to the orthogonality condition derived in (3.18).

It is of great interest, however, to analyze the evolution of the coefficients of the filter in time. We will derive a first order study of the *convergence in the mean*, to verify whether $\mathbf{w}^{(k)} \rightarrow \mathbf{w}_{MMSE}$, when $k \rightarrow \infty$.

Definition (Convergence in the mean): We say that a sequence $y(k)$ converges in the mean towards $d(k)$, if the mean of both sequences exist and

$$\lim_{k \rightarrow \infty} E\{e(k)\} = \lim_{k \rightarrow \infty} E\{d(k) - y(k)\} = 0. \quad (3.27)$$

Theorem 3.2 (Convergence of the CLMS algorithm)

Given a CLMS algorithm that designs a filter $\mathbf{w} \in \mathbb{C}^m$ to estimate a desired sequence $d(k)$ from an input sequence $x(k)$, we can ensure that it converges in the mean to the optimal solution \mathbf{w}_{MMSE} if the step size $\mu^{(k)}$ satisfies the condition

$$\mu^{(k)} < \frac{2}{mE\{|x(k)|^2\}}. \quad (3.28)$$

Proof: For convenience, the analysis will be based upon a variation of the form (3.1), given by

$$y(k) = \mathbf{x}^H(k) \mathbf{w}, \quad (3.29)$$

which leads to the conjugate cogradient

$$\nabla_{\mathbf{w}^*} J_{LMS}(\mathbf{w}) = -\mathbf{x}(k)e(k). \quad (3.30)$$

However, this is only because of the mathematical tractability. Suppose that the desired sequence $d(k)$ can be expressed as

$$d(k) = \mathbf{x}^H(k) \mathbf{w}_{MMSE} + q(k), \quad (3.31)$$

where $q(k)$ is a complex Additive White Gaussian Noise, with zero mean and variance σ_q^2 . This structure reflects a part of the desired sequence that the model can explain ($\mathbf{x}^H(k) \mathbf{w}_{MMSE}$), and another part that the model cannot explain ($q(k)$). Substituting (3.29) and (3.31) in (3.4), the error sequence is

$$e(k) = \mathbf{x}^H(k) \mathbf{w}_{MMSE} + q(k) - \mathbf{x}^H(k) \mathbf{w}^{(k)}, \quad (3.32)$$

and the update (3.20) results,

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} + \mu^{(k)} \mathbf{x}(k) \mathbf{x}^H(k) \mathbf{w}_{MMSE} \\ &\quad - \mu^{(k)} \mathbf{x}(k) \mathbf{x}^H(k) \mathbf{w}^{(k)} + \mu^{(k)} \mathbf{x}(k) q(k). \end{aligned} \quad (3.33)$$

If we now define the vector of errors in the coefficients of the filter, $\Delta \mathbf{w}^{(k)} = \mathbf{w}^{(k)} - \mathbf{w}_{MMSE}$ and subtract \mathbf{w}_{MMSE} from both sides of (3.33),

$$\Delta \mathbf{w}^{(k+1)} = \Delta \mathbf{w}^{(k)} - \mu^{(k)} \mathbf{x}(k) \mathbf{x}^H(k) \Delta \mathbf{w}^{(k)} + \mu^{(k)} \mathbf{x}(k) q(k). \quad (3.34)$$

By taking statistical expectation on both sides,

$$E\{\Delta \mathbf{w}^{(k+1)}\} = \left(I - \mu^{(k)} E\{\mathbf{x}(k) \mathbf{x}^H(k)\}\right) E\{\Delta \mathbf{w}^{(k)}\} + \mu^{(k)} E\{\mathbf{x}(k) q(k)\}. \quad (3.35)$$

By assuming independence, we can identify the covariance of the input (3.8) and vanish the last cross-covariance term,

$$E\{\Delta \mathbf{w}^{(k+1)}\} = \left(I - \mu^{(k)} \mathbf{R}_{\mathbf{x}}\right) E\{\Delta \mathbf{w}^{(k)}\}. \quad (3.36)$$

Since $\mathbf{x}(k)$ is WSS, the covariance matrix $\mathbf{R}_{\mathbf{x}}$ is Hermitian and *positive semidefinite*⁴. Thus, it can be decomposed by using the *Eigenvalue*

⁴ This means that $\mathbf{R}_{\mathbf{x}}$ satisfies $\mathbf{a}^H \mathbf{R}_{\mathbf{x}} \mathbf{a} \geq 0$, for all $\mathbf{a} \in \mathbb{C}^m$.

Decomposition (EVD) into a diagonal matrix $\mathbf{\Lambda}$ and a unitary transformation \mathbf{Q} , such as

$$\mathbf{R}_x = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H, \quad (3.37)$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_m \end{bmatrix}, \quad \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0 \quad (3.38)$$

is a diagonal matrix comprising the real and positive *eigenvalues* λ_i of \mathbf{R}_x , sorted in descending order, and \mathbf{Q} is a unitary matrix that contains the m corresponding eigenvectors. See (Golub, 1996) for a broad treatment of this and others algebra topics.

We can rewrite (3.36) as

$$\begin{aligned} E\{\Delta\mathbf{w}^{(k+1)}\} &= (I - \mu^{(k)}\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H)E\{\Delta\mathbf{w}^{(k)}\} \\ &= \mathbf{Q}(I - \mu^{(k)}\mathbf{\Lambda})\mathbf{Q}^H E\{\Delta\mathbf{w}^{(k)}\}. \end{aligned} \quad (3.39)$$

The rotation by the matrix \mathbf{Q} allows us to express the *modes of convergence* exclusively in terms of the corresponding eigenvalues of \mathbf{R}_x . If we define

$$\Delta\mathbf{w}_Q^{(k)} = \mathbf{Q}^H \Delta\mathbf{w}^{(k)}, \quad (3.40)$$

then

$$E\{\Delta\mathbf{w}_Q^{(k+1)}\} = (I - \mu^{(k)}\mathbf{\Lambda})E\{\Delta\mathbf{w}_Q^{(k)}\}. \quad (3.41)$$

Every element of $\Delta\mathbf{w}_Q^{(k)}$ evolves independently, because the multiplication of $\Delta\mathbf{w}^{(k)}$ by the unitary matrix \mathbf{Q} decouples the evolution of its coefficients. If one wants to converge in the mean, one must ensure that the expected value becomes smaller with each iteration,

$$E\{\Delta\mathbf{w}_Q^{(k+1)}\} < E\{\Delta\mathbf{w}_Q^{(k)}\}. \quad (3.42)$$

By looking at (3.41), the condition (3.42) for each mode of convergence is satisfied when

$$\left|1 - \mu^{(k)}\lambda_i\right| < 1. \quad (3.43)$$

The largest eigenvalue, $\lambda_1 = \lambda_{MAX} = \max_i \{\lambda_i\}$, controls if the fastest mode converges or not, so the condition for the convergence in the mean becomes:

$$\mu^{(k)} < \frac{2}{\lambda_{MAX}}. \quad (3.44)$$

The largest eigenvalue is always smaller than the *trace*⁵ of the covariance matrix, so

$$\lambda_{MAX} \leq Tr\{\mathbf{R}_x\}. \quad (3.45)$$

And by the definition (3.8), we know that the trace of a covariance matrix can be computed directly from the input signal,

$$Tr\{\mathbf{R}_x\} = mE\{|x(k)|^2\}. \quad (3.46)$$

■

Another type of second order convergence study can be derived. Once the algorithm starts to converge in the mean, the filter coefficients begin to fluctuate around the optimal values. This is due to the stochastic behavior of the algorithm. The *convergence in the mean square* establishes conditions to avoid this problem, and a detailed derivation can be found on (Mandic, 2009).

3.4 Blind Source Separation and Blind Source Extraction in the Complex Domain

Once we have seen some supervised applications, it is time to deal with *blind* ones, that is, scenarios where we do not have access to the training data nor the mixture. In these cases, we have to exploit some structural or statistical properties of the signals involved in order to achieve our goal. *Blind Source Separation (BSS)* has been one of the most active topics in the signal processing literature in the past years. For a unified treatment of this research field, see (Chichocki, 2002), (Comon, 2010), (Cardoso, 1998) and (Hyvärinen, 2001).

Consider a scenario where a number of n signals are emitted by some physical sources. Assume that there are m sensors that receive these signals after a linear transformation, defined by the matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$. The *sources* are confined into

⁵ The trace of a matrix is defined as the sum of the diagonal elements, and that coincides with the sum of the eigenvalues: $Tr\{\mathbf{R}\} = \sum_i r_{ii} = \sum_i \lambda_i$

the complex vector of random processes $\mathbf{s}(k) \in \mathbb{C}^n$, and the *observations* are confined into $\mathbf{x}(k) \in \mathbb{C}^m$. So, the *noiseless linear mixing model* results,

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k). \quad (3.47)$$

For the sake of simplicity, we will assume initially that the number of sources equals the number of observations ($n = m$). In a BSS problem, we want to recover the original sources $\mathbf{s}(k)$ only from the observations $\mathbf{x}(k)$, without the knowledge of the mixing matrix \mathbf{A} . The only information that we may have available is some statistical or geometrical properties of the sources.

Usually, the signal separation is done by designing a *separation matrix* $\mathbf{B} \in \mathbb{C}^{n \times m}$ that recovers an estimation of the original sources $\hat{\mathbf{s}}(k) \in \mathbb{C}^n$, when applied to the observations $\mathbf{x}(k)$,

$$\hat{\mathbf{s}}(k) = \mathbf{B}^H \mathbf{x}(k) = \mathbf{B}^H \mathbf{A}\mathbf{s}(k). \quad (3.48)$$

Of course, if we suppose that \mathbf{A} is invertible and we know all of its coefficients, a_{ij} , the solution of the separation problem is the inverse of the mixing matrix:

$$\mathbf{B}_{opt}^H = \mathbf{A}^{-1}. \quad (3.49)$$

But as we have said, in a blind problem we do not know the value of a_{ij} . The other solution is to estimate directly the coefficients b_{ij} of the separation matrix, and most of the proposed methods (see (Chichocki, 2002)) are based upon this idea.

The problem of recovering only one of the sources is called *Blind Source Extraction (BSE)*. Usually, it is easier to extract a source than to separate the whole set. A BSS problem can be solved by extracting the sources sequentially, one by one, and doing a *deflation* after each extraction (Delfosse, 1995). Alternatively, it can be solved by a joint separation, recovering all the sources simultaneously like in the *Joint Approximate Diagonalization of Eigenmatrices (JADE)* algorithm (Cardoso, 1997).

As we only observe $\mathbf{x}(k)$ and we want to recover $\mathbf{s}(k)$, there are virtually infinite solutions. We need to assume some property of the sources to constraint the possible solutions. These properties go from very loose to very tight, like the statistical independence (Comon, 1994), the uncorrelatedness (Jolliffe, 2002), the sparsity (Li, 2003) or the non-negativity (Cichocki, 2009). We overview two popular approaches that solve this BSS problem.

3.4.1 Principal Component Analysis, decorrelation and whitening

The method called *Principal Component Analysis (PCA)* is based upon an idea that has been known long ago (Hotelling, 1935). The main assumption on the sources is that there is no *cross-correlation* between them, that is,

$$r_{ij} = E\{s_i(k)s_j^*(k)\} = \begin{cases} \sigma_i^2, & i = j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, \dots, n, \quad (3.50)$$

where σ_i^2 is the variance of the i -th source. We have supposed that all the signals involved have zero mean. If this is not the case, it is advisable to subtract the mean from all the sequences,

$$\mathbf{x}(k) \leftarrow \mathbf{x}(k) - E\{\mathbf{x}(k)\}. \quad (3.51)$$

This method is closely related to the *Karhunen-Loève transform* or the *Hotelling transform*. The purpose is to find a set of variables $\mathbf{s}(k)$ with less redundancy than the observations $\mathbf{x}(k)$. The measure of redundancy will be the cross-correlation.

The main advantage of PCA is that it is based only on second-order statistics. The main drawback is that usually, the uncorrelatedness assumption is a very poor model of the physics underlying the mixture model, so the mixture is not fully determined (there are n^2 unknowns and $n(n-1)/2$ equations in the problem). Nevertheless, PCA is widely used in practical applications as a first step to more refined procedures (Hyvärinen, 2001), (Tome, 2004), (Zarzoso, 2012).

We show here the process of *whitening*, which consists on a *decorrelation* (or PCA) followed by a scaling (Hyvärinen, 2001). Suppose that all the sources have unit variance, so the covariance matrix of the sources must be the identity matrix,

$$\mathbf{R}_s = E\{\mathbf{s}(k)\mathbf{s}^H(k)\} = \mathbf{I} \in \mathbb{C}^{n \times n}. \quad (3.52)$$

The key behind the decorrelation procedure is the following. We know that the observations $\mathbf{x}(k)$ are the result of a linear combination of the sources like in (3.47), so they will be more correlated than the original signals. That is, the correlation matrix \mathbf{R}_x will have nonzero elements outside the main diagonal.

Imagine now that we can derive a set of *whitened sequences* $\mathbf{z}(k)$, after multiplying the observations $\mathbf{x}(k)$ by a separation matrix $\mathbf{W} \in \mathbb{C}^{n \times m}$,

$$\mathbf{z}(k) = \mathbf{W}\mathbf{x}(k). \quad (3.53)$$

These set of sequences will have a covariance matrix \mathbf{R}_z very similar to the identity matrix. Our goal is to design the matrix \mathbf{W} to minimize the cross-correlation between elements of $\mathbf{z}(k)$,

$$\mathbf{R}_z = E\{\mathbf{z}(k)\mathbf{z}^H(k)\} \approx \mathbf{I} \in \mathbb{C}^{n \times n}. \quad (3.54)$$

The output sequences must be a good estimate of the original sources, $\mathbf{z}(k) = \hat{\mathbf{s}}(k)$, given the maximum decorrelation criterion. The value of the separation matrix \mathbf{W} is given by the following theorem.

Theorem 3.3 (Whitening)

Given the linear mixing model (3.47) and the separation (3.48), the desired separation matrix $\mathbf{W} \in \mathbb{C}^{n \times n}$ is given by

$$\mathbf{W} = \mathbf{\Sigma}^{-1/2} \mathbf{Q}^H, \quad (3.55)$$

where $\mathbf{\Sigma} \in \mathbb{C}^{n \times n}$ is the matrix containing the eigenvalues of \mathbf{R}_x in its diagonal, and $\mathbf{Q} \in \mathbb{C}^{n \times n}$ is the unitary matrix whose columns are the corresponding eigenvectors of \mathbf{R}_x . That is, \mathbf{R}_x admits an eigenvalue decomposition

$$\mathbf{R}_x = \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^H. \quad (3.56)$$

The resulting sequences $\mathbf{z}(k)$ given by (3.53) are mutually uncorrelated and with unit variance (white).

Proof: We start by substituting (3.53) in the correlation matrix of $\mathbf{z}(k)$,

$$\mathbf{R}_z = E\{\mathbf{z}(k)\mathbf{z}^H(k)\} = \mathbf{W}E\{\mathbf{x}(k)\mathbf{x}^H(k)\}\mathbf{W}^H = \mathbf{W}\mathbf{R}_x\mathbf{W}^H. \quad (3.57)$$

Using the eigenvalue decomposition of \mathbf{R}_x ,

$$\mathbf{R}_z = \mathbf{W}\mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^H\mathbf{W}^H = (\mathbf{W}\mathbf{Q}\mathbf{\Sigma}^{1/2})(\mathbf{\Sigma}^{1/2}\mathbf{Q}^H\mathbf{W}^H), \quad (3.58)$$

where $\mathbf{\Sigma}^{1/2}$ is a diagonal matrix whose elements are the square roots of the elements in the diagonal of $\mathbf{\Sigma}$. By substituting (3.55) in (3.58), and by noting that \mathbf{Q} is unitary and $\mathbf{\Sigma}$ is a diagonal matrix with real, positive elements,

$$\mathbf{R}_z = (\mathbf{\Sigma}^{-1/2}\mathbf{Q}^H\mathbf{Q}\mathbf{\Sigma}^{1/2})(\mathbf{\Sigma}^{1/2}\mathbf{Q}^H\mathbf{Q}\mathbf{\Sigma}^{-1/2}) = (\mathbf{\Sigma}^{-1/2}\mathbf{\Sigma}^{1/2})(\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{-1/2}) = \mathbf{I}. \quad (3.59)$$

So we have verified that $\mathbf{z}(k)$ has been whitened⁶. ■

The matrix \mathbf{Q} is responsible of the decorrelation, and the matrix $\mathbf{\Sigma}^{-1/2}$ is responsible for the scaling the sequences, so the output has unit power,

$$E\{z_i(k)z_i^*(k)\} = \sigma_i^2 = 1, \quad i = 1, \dots, n. \quad (3.60)$$

Note that, if we obviate the matrix $\mathbf{\Sigma}^{-1/2}$, we get a uncorrelated set of sequences but without unit power, called *principal components* (Abdi, 2010).

Theorem 3.3 is a powerful result that has been widely used in statistics, communications, image and audio processing and many other applications. However, it is only based on second-order statistics and lacks to separate signals with more intricate relations. Anyway, this useful tool is used as a preprocessing step in several applications to prepare the sequences for an algorithm.

3.4.2 Independent Component Analysis

The next step of BSS is to consider that the desired sources are not only uncorrelated. In addition, we will impose the stronger assumption of *mutual independence* (Gray, 2010). This means that not only the second order cross-moments vanishes, but the *joint probability density function* of each two sources satisfies

$$p(z_i(k))p(z_j(k)) = p(z_i(k), z_j(k)), \quad i, j = 1, \dots, n \quad (i \neq j), \quad (3.61)$$

where $p(z_i(k), z_j(k))$ is the joint probability density function of $z_i(k)$ and $z_j(k)$. Independence is a very strong condition, and implies uncorrelatedness. The opposite is not true, unless the variables involved are *jointly Gaussian* (see (Melnick, 1982)).

If one tries to separate a set of sources that are independent, by using a correlation criterion like in PCA, it fails. However, PCA or whitening usually are the halfway to ICA, because at least they decorrelate the sequences and prepares them for further processing. Nevertheless, we need more than decorrelation to achieve our goal.

The *Central Limit Theorem* is a classical result in probability (see (Rice, 1995) for an introduction), which states that the distribution of a sum of independent

⁶ Note that this solution is by no means the only unique whitening matrix. It is easy to see that any matrix \mathbf{WU} , with \mathbf{U} unitary, is also a whitening matrix.

random sequences tends toward a Gaussian distribution, under very loose conditions. Intuitively, this means that the linear combination of independent sources usually has a distribution that is closer to Gaussian than any of the original sources. As they lose independence due to the combining, they gain Gaussianity.

The simplest approach is to use a cost function that measures non-Gaussianity in some practical way. There are several works that propose new contrasts for this problem (see (Novey, 2008) or (Novey, 2006) for examples), and we present here one of the more intuitive.

The *kurtosis* is a fourth-order statistical moment, and can be defined in many ways. We use a simple definition from (Therrien, 1992), that for a proper zero mean complex white random process $y(k)$ results

$$\text{kurt}(y(k)) \triangleq E\{|y(k)|^4\} - 2E\{|y(k)|^2\}^2. \quad (3.62)$$

Assume that the sources have unit power, so $E\{|y(k)|^2\} = \sigma_y^2 = 1$ and the kurtosis results

$$\text{kurt}(y(k)) = E\{|y(k)|^4\} - 2 = E\{(y(k)y^*(k))^2\} - 2. \quad (3.63)$$

Intuitively, kurtosis measures long tails or peakedness in the shape of the probability density functions. A high positive kurtosis means that the distribution is very peaky, like a *Laplacian*. These distributions are called *supergaussians*.

A very negative kurtosis means that the distribution has small tails, like a *uniform*. These distributions are called *subgaussians*. In the Figure 3.4, two examples of these kinds of distributions are shown. The uniform distribution (red), as well as other bounded ones, is subgaussian and has negative kurtosis. On the other hand, the Laplacian distribution (green), as well as other peaky ones, is supergaussian and has positive kurtosis.

The Gaussian distribution has zero kurtosis, and thus the absolute value of the kurtosis is a good measure of non-Gaussianity.

$$J_{ICA}(\mathbf{b}) = |\text{kurt}(y(k))|, \quad (3.64)$$

where

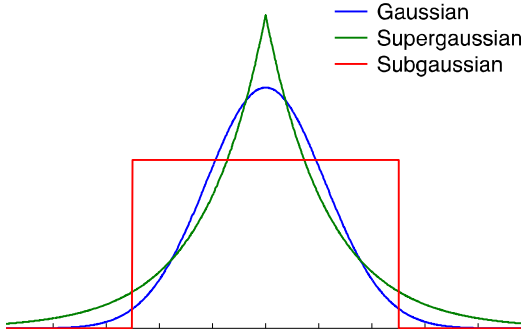


Figure 3.4 – Supergaussian and subgaussian distributions. A Gaussian distribution is plotted for comparison.

$$y(k) = \mathbf{b}^H \mathbf{z}(k) = \hat{s}_1(k) \quad (3.65)$$

is the extraction model for the first (without loss of generality) source, and \mathbf{b} is the *extraction vector*. Note that this time the observations are *whitened observations* $\mathbf{z}(k)$, taken from the result of a whitening procedure. The Figure 3.5 shows a diagram of the complete system.

We just need to compute the conjugate cogradient of (3.64) to develop a steepest descent algorithm, like in Section 2.4.3. The Theorem 3.4 shows the resulting expression for the update equation.

Theorem 3.4 (ICA by maximization of the non-Gaussianity)

Given the model defined in (3.65), the update equation for a steepest descent algorithm that solves the ICA extraction problem by maximizing the non-Gaussianity, is

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + 2\mu^{(k)} \text{sign}\left(\text{kurt}(y(k))\right) E\left\{\left(y^*(k)\right)^2 y(k) \mathbf{z}(k)\right\}. \quad (3.66)$$

After convergence, an algorithm with this update will extract successfully the source with the highest absolute kurtosis.

Proof: The conjugate cogradient of (3.64) is

$$\nabla_{\mathbf{b}^*} \left(J_{\text{ICA}}(\mathbf{b}) \right) = \text{sign}\left(\text{kurt}(y(k))\right) \nabla_{\mathbf{b}^*} \left(\text{kurt}(y(k)) \right). \quad (3.67)$$

Note that, by using (3.65), we have the following derivatives,

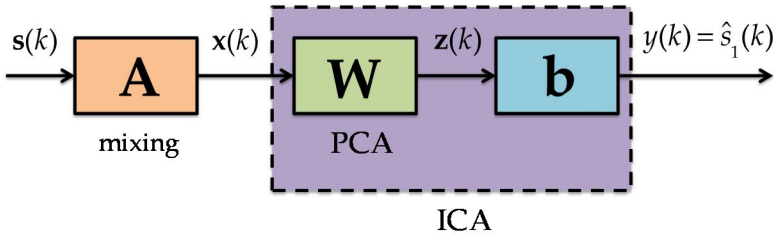


Figure 3.5 - Diagram of the linear mixing and extraction method for Independent Component Analysis. It is important to whiten the observations so the cost function is more tractable. Also, this improves the convergence and the robustness of the algorithms derived.

$$\begin{aligned} \nabla_{\mathbf{b}^*} (y(k)) &= \mathbf{z}(k) & \nabla_{\mathbf{b}^*} (y^*(k)) &= \mathbf{0} \\ \nabla_{\mathbf{b}} (y(k)) &= \mathbf{0} & \nabla_{\mathbf{b}} (y^*(k)) &= \mathbf{z}^*(k). \end{aligned} \quad (3.68)$$

So, the conjugate cogradient of the kurtosis (3.63) is

$$\begin{aligned} \nabla_{\mathbf{b}^*} (kurt(y(k))) &= E \left\{ \nabla_{\mathbf{b}^*} \left((y(k)y^*(k))^2 \right) \right\} \\ &= E \left\{ \frac{\partial (y(k)y^*(k))^2}{\partial y(k)} \nabla_{\mathbf{b}^*} (y(k)) \right\} \\ &= 2E \left\{ (y^*(k))^2 y(k) \mathbf{z}(k) \right\}. \end{aligned} \quad (3.69)$$

And the update in the steepest descent algorithm results in

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + 2\mu^{(k)} \text{sign}(kurt(y(k))) E \left\{ (y^*(k))^2 y(k) \mathbf{z}(k) \right\}. \quad (3.70)$$

■

This is only for the extraction of one source. If we want to separate the whole set, we can extend the algorithm to build a separation matrix $\mathbf{B} = [\mathbf{b}_1 \ \cdots \ \mathbf{b}_n]$, whose columns are the extraction vectors. This is done by using a Gram-Schmidt procedure (Herstein, 1964) after the building of each \mathbf{b}_i ,

$$\mathbf{b}_i \leftarrow \mathbf{b}_i - \sum_{r=1}^{i-1} (\mathbf{b}_i^H \mathbf{b}_r) \mathbf{b}_r, \quad \mathbf{b}_i \leftarrow \frac{\mathbf{b}_i}{\|\mathbf{b}_i\|}. \quad (3.71)$$

And this sequential process of extraction and projection can be used to separate signals blindly, with independence of the method used (see (Durán-Díaz, 2011) for a discussion). Note that, even if the separation is successful, there are two ambiguities remaining on the solution:

- We cannot determine the scale of the independent components. This is because the mixing matrix are as unknown as the sources, so a coefficient α_i in a source $s_i(k)$ can be cancelled by its counterpart $1/\alpha_i$ in the columns of the mixing matrix \mathbf{a}_i . This is why the output is forced to have unit variance and the extraction vector is normalized.
- We cannot determine the order of the independent components. The components are extracted in an order defined by the absolute kurtosis, and due to the commutative property of addition, we cannot know which one was the order of the original sources.

Fortunately, these problems have been solved for most of the practical scenarios. Usually, they are circumvented by making certain assumptions, or by further post-processing on the output sequences like in (Sarmiento, 2012), (Zarzoso, 2010), or (Durán-Díaz, 2012).

By looking at the results of this section, one might think that, to pass from the real to the complex world, one only has to use the conjugate operator $(\cdot)^*$ in some of the variables involved. That is because, until now, we have assumed a certain structure on the statistical distribution that is valid only in some situations. We are talking about circularity.

However, when the statistical structure of the variables is not so simple, the results derived from this naive simplification causes algorithms with poor performance. Fortunately, during the last decade there have been very successful advances on the modeling of the more intricate relations between the conjugate parts of a complex variable.

3.5 Widely Linear Wiener Filtering

In this section, we move on to the case of widely linear models, aiming to exploit the additional second-order information contained in the pseudo-covariance matrix. We derive the Widely Linear Wiener solution to the MMSE criterion by using the information contained in both the sequence and the conjugate sequence. This problem was first addressed in (Picinbono, 1995).

Imagine a problem like the one presented in Section 3.2, where we want to estimate a desired sequence $d(k)$ with the result of passing an input sequence $x(k)$ and its conjugate $x^*(k)$ through two complex linear filters, called \mathbf{h} and

$\tilde{\mathbf{h}}$, and adding their outputs. We obtain

$$y(k) = \sum_{j=1}^m h_j^* x(k-j+1) + \tilde{h}_j x^*(k-j+1) = \mathbf{h}^H \mathbf{x}(k) + \tilde{\mathbf{h}}^H \mathbf{x}^*(k), \quad (3.72)$$

where $\mathbf{x}(k)$ and $\mathbf{x}^*(k)$ store delayed copies of the corresponding sequences, like in (3.2).

We aim to minimize again the error in the MSE sense, that is

$$\mathbf{h}_{WLMSE} = \arg \left\{ \min_{\mathbf{h}} E \left\{ |e(k)|^2 \right\} \right\}, \quad (3.73)$$

$$\tilde{\mathbf{h}}_{WLMSE} = \arg \left\{ \min_{\tilde{\mathbf{h}}} E \left\{ |e(k)|^2 \right\} \right\}, \quad (3.74)$$

with $e(k) = d(k) - y(k)$.

The new cost function is called the *Widely Linear Mean Square Error (WLMSE)*. See (Pedzisz, 2007) for a discussion on the linearity of the moments used in this criterion.

The solutions of (3.73) and (3.74) are given by deriving a similar reasoning that in Theorem 3.1.

Theorem 3.5 (Widely Linear Solution to the MMSE criterion)

Given a *widely linear filter*, represented by the pair $\mathbf{h}, \tilde{\mathbf{h}}$, that estimates a desired sequence $d(k)$ using a linear combination of input samples $x(k)$, the solution to the MMSE criterion is:

$$\mathbf{h} = \left(\mathbf{R}_{\mathbf{x}} - \tilde{\mathbf{R}}_{\mathbf{x}} \left(\mathbf{R}_{\mathbf{x}}^{-1} \right)^* \tilde{\mathbf{R}}_{\mathbf{x}}^* \right)^{-1} \left(\mathbf{r}_{\mathbf{x}d} - \tilde{\mathbf{R}}_{\mathbf{x}} \left(\mathbf{R}_{\mathbf{x}}^{-1} \right)^* \tilde{\mathbf{r}}_{\mathbf{x}d}^* \right), \quad (3.75)$$

$$\tilde{\mathbf{h}} = \left(\mathbf{R}_{\mathbf{x}}^* - \tilde{\mathbf{R}}_{\mathbf{x}}^* \mathbf{R}_{\mathbf{x}}^{-1} \tilde{\mathbf{R}}_{\mathbf{x}} \right)^{-1} \left(\tilde{\mathbf{r}}_{\mathbf{x}d}^* - \tilde{\mathbf{R}}_{\mathbf{x}}^* \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}d} \right), \quad (3.76)$$

where

$$\mathbf{R}_{\mathbf{x}} = E \left\{ \mathbf{x}(k) \mathbf{x}^H(k) \right\}, \quad (3.77)$$

$$\tilde{\mathbf{R}}_{\mathbf{x}} = E \left\{ \mathbf{x}(k) \mathbf{x}^T(k) \right\}, \quad (3.78)$$

are the covariance and the pseudo-covariance of the input, and

$$\mathbf{r}_{\mathbf{x}d} = E \left\{ \mathbf{x}(k) d^*(k) \right\}, \quad (3.79)$$

$$\tilde{\mathbf{r}}_{xd} = E\{\mathbf{x}(k)d(k)\}, \quad (3.80)$$

are the cross-covariance and the cross-pseudo-covariance between the input and the desired sequence.

Proof: We proceed as in the proof of the Wiener solution for the MMSE criterion. Computing the gradient of the cost function, we obtain:

$$\nabla_{\mathbf{h}} E\{|e(k)|^2\} = E\left\{\nabla_{\mathbf{h}} \cdot \left((d(k) - y(k))(d^*(k) - y^*(k)) \right)\right\}. \quad (3.81)$$

Using the definition of $y(t)$ from (3.72), and ignoring all the terms that do not depend on \mathbf{h}^* ,

$$\begin{aligned} \nabla_{\mathbf{h}} E\{|e(k)|^2\} = & -E\left\{\nabla_{\mathbf{h}} \cdot (\mathbf{h}^H \mathbf{x}(k)d^*(k))\right\} + E\left\{\nabla_{\mathbf{h}} \cdot (\mathbf{h}^H \mathbf{x}(k)\mathbf{h}^T \mathbf{x}^*(k))\right\} \\ & + E\left\{\nabla_{\mathbf{h}} \cdot (\mathbf{h}^H \mathbf{x}(k)\tilde{\mathbf{h}}^T \mathbf{x}(k))\right\}. \end{aligned} \quad (3.82)$$

Computing the gradients,

$$\nabla_{\mathbf{h}} E\{|e(k)|^2\} = -E\{\mathbf{x}(k)d^*(k)\} + E\{\mathbf{x}(k)\mathbf{x}^H(k)\}\mathbf{h} + E\{\mathbf{x}(k)\mathbf{x}^T(k)\}\tilde{\mathbf{h}}. \quad (3.83)$$

By identifying terms in the definitions (3.77)-(3.80), and by vanishing the gradient,

$$\mathbf{r}_{xd} = \mathbf{R}_x \mathbf{h} + \tilde{\mathbf{R}}_x \tilde{\mathbf{h}}. \quad (3.84)$$

If we apply the same procedure but for $\tilde{\mathbf{h}}$, we obtain

$$\begin{aligned} \nabla_{\tilde{\mathbf{h}}} E\{|e(k)|^2\} = & -E\{\mathbf{x}^*(k)d^*(k)\} + E\{\mathbf{x}^*(k)\mathbf{x}^H(k)\}\mathbf{h} \\ & + E\{\mathbf{x}^*(k)\mathbf{x}^T(k)\}\tilde{\mathbf{h}}. \end{aligned} \quad (3.85)$$

So, the second equation for the widely linear model is

$$\tilde{\mathbf{r}}_{xd}^* = \tilde{\mathbf{R}}_x^* \mathbf{h} + \mathbf{R}_x^* \tilde{\mathbf{h}}. \quad (3.86)$$

By solving the linear system of matrix equations formed by (3.84) and (3.86), we obtain the MMSE solution for the widely linear filter as seen in (3.75) and (3.76). ■

By using a widely linear filter, we are exploiting additional information. It is well known that using the covariance, we are losing the rich information in the phase of the second order statistics. By adding the pseudo-covariance, we recover that information and use it to design a more accurate filter.

Note that we can reformulate the widely linear filter equation (3.72) by using an augmented representation like in Section 2.5.3. In fact,

$$y(k) = \underline{\mathbf{h}}^H \underline{\mathbf{x}}(k), \quad (3.87)$$

where

$$\underline{\mathbf{h}} = \begin{bmatrix} \mathbf{h} \\ \tilde{\mathbf{h}} \end{bmatrix} \in \mathbb{C}^{2m}, \quad (3.88)$$

and

$$\underline{\mathbf{x}}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}^*(k) \end{bmatrix} \in \mathbb{C}^{2m} \quad (3.89)$$

are the augmented representations of the filter and the input, respectively. Note that $\underline{\mathbf{h}}$ is a relaxation of the notation, because it does not have the structure of an augmented vector. Anyway, the augmented representation provides an elegant framework to work with the richer statistics of improper random vectors.

Corollary to Theorem 3.5

When the pseudo-covariance matrix $\tilde{\mathbf{R}}_{\mathbf{x}}$ and the cross-pseudo-covariance vector $\tilde{\mathbf{r}}_{\mathbf{x}d}$ vanish, the filter \mathbf{h} of the WLMSE solution reduces to the Wiener solution \mathbf{w} for the MMSE criterion.

$$\mathbf{h} = \mathbf{R}_{\mathbf{x}}^{-1} \mathbf{r}_{\mathbf{x}d}, \quad (3.90)$$

$$\tilde{\mathbf{h}} = \mathbf{0}. \quad (3.91)$$

The filter $\tilde{\mathbf{h}}$, which works on the conjugate part of $\mathbf{x}(k)$, vanishes. This phenomenon occurs when the signals are jointly circular.

Proof: To show this, simply substitute $\tilde{\mathbf{R}}_{\mathbf{x}} = \mathbf{0}$ and $\tilde{\mathbf{r}}_{\mathbf{x}d} = \mathbf{0}$ on the solutions to the WLMSE problem, (3.75) and (3.76). ■

We have seen that, when the signals are improper, the solution derived from the minimization of the WLMSE criterion is slightly different from the one derived

from the MMSE criterion. ¿How can we measure this difference?

The most logical quantity is the mismatch between the mean square errors using both estimators. The mean square error of the linear solution is given by (3.16),

$$e_{MSE}^2 = E \left\{ \left| e(k) \right|^2 \right\} = \sigma_d^2 - \mathbf{r}_{xd}^H \mathbf{R}_x^{-1} \mathbf{r}_{xd}. \quad (3.92)$$

And now, the mean square error of the widely linear solution is given by

$$e_{WLMSE}^2 = \sigma_d^2 - \left(\mathbf{h}^H \mathbf{r}_{xd} + \tilde{\mathbf{h}}^H \tilde{\mathbf{r}}_{xd}^* \right). \quad (3.93)$$

The *advantage of the widely linear MSE* over the linear MSE is characterized by the quantity $\delta e^2 = e_{WLMSE}^2 - e_{MSE}^2$, and can be expressed as

$$\delta e^2 = \left(\tilde{\mathbf{r}}_{xd}^* - \tilde{\mathbf{R}}_x^* \mathbf{R}_x^{-1} \mathbf{r}_{xd} \right)^H \left(\mathbf{R}_x^* - \tilde{\mathbf{R}}_x^* \mathbf{R}_x^{-1} \tilde{\mathbf{R}}_x \right)^{-1} \left(\tilde{\mathbf{r}}_{xd}^* - \tilde{\mathbf{R}}_x^* \mathbf{R}_x^{-1} \mathbf{r}_{xd} \right). \quad (3.94)$$

Note that $\delta e^2 \geq 0$, because $\left(\mathbf{R}_x^* - \tilde{\mathbf{R}}_x^* \left(\mathbf{R}_x^{-1} \right) \tilde{\mathbf{R}}_x \right)$ is positive definite as stated in (Picinbono, 1996). This means that the advantage vanishes only when

$$\tilde{\mathbf{r}}_{xd}^* = \tilde{\mathbf{R}}_x^* \left(\mathbf{R}_x^{-1} \right) \mathbf{r}_{xd}. \quad (3.95)$$

That is when $\tilde{\mathbf{h}} = \mathbf{0}$, and we are again in the linear model. See (Pedzisz, 2007) for a further discussion on the advantages of widely linear filter.

Example 3.1 *Let us see an application where the widely linear modeling takes advantage of the additional information confined in the pseudo-covariance. In particular, the objective will be to separate two sources from a linear mixture $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$, using a supervised criterion like the minimum MSE.*

In this experiment⁷, we used a set of two sources coming from two speakers, and two observations from a synthetic linear mixture in the presence of noise. Obviating details of the algorithm, we performed a local second-order analysis on the speech to conform the covariance and pseudo-covariance matrices.

It is well known that the speech is a non-stationary signal. This means that its statistical properties changes over time, so usually engineers work in the frequency domain by transforming speech signals by means of the Short Time Fourier Transform (STFT), obtaining complex-valued sequences.

⁷ This experiment was developed during the short stay of the proponent at the Imperial College London with the research group of Prof. Danilo Mandic, at the Electronic and Electrical Engineering Department. The study showed an original application of widely linear estimation.

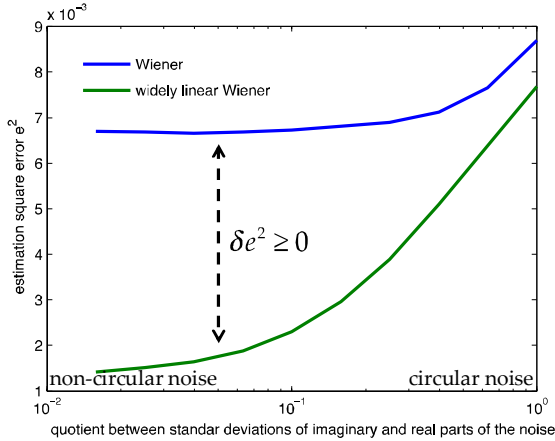


Figure 3.6 - This figure shows the comparison between the estimation square error of the Wiener (blue) and the Widely Linear Wiener (green) solutions to the speech separation problem, in the presence of additive noise.

In (Picinbono, 1994), it was demonstrated that STFT coefficients of stationary processes are complex circular. Following that reasoning, (Benesty, 2010) proposed that the STFT coefficients of non-stationary signals were, in general, complex non-circular.

Thus, we can exploit the rich improper structure of the sources and the noise to design a widely linear extraction vector that performs better than the linear approach. The model used is the widely linear extension of the separation model (3.48),

$$\hat{\mathbf{s}}(k) = \mathbf{B}^H \mathbf{x}(k) + \tilde{\mathbf{B}}^H \mathbf{x}^*(k). \quad (3.96)$$

Figure 3.6 compares the error of the linear and the widely linear estimation, corroborating what (3.94) declares. As stated before, it is always positive, meaning that the error in the widely linear case is always lower than in the linear case. The abscise axis shows the ratio between the standard deviations of the imaginary and the real part of the noise. Thus, the left part of the plot corresponds to a non-circular noise, while the right part shows a scenario with circular noise.

Figure 3.7 shows the comparison of the estimation results of a sample frame of some milliseconds of speech. It can be seen that the widely linear estimate is closer to the original source than the linear one. The blue line is the temporal stamp of one sub-band of the original source. The green line is the estimation, using both approximations. The difference between the desired and the estimated sequence is the red line, namely the estimation error.

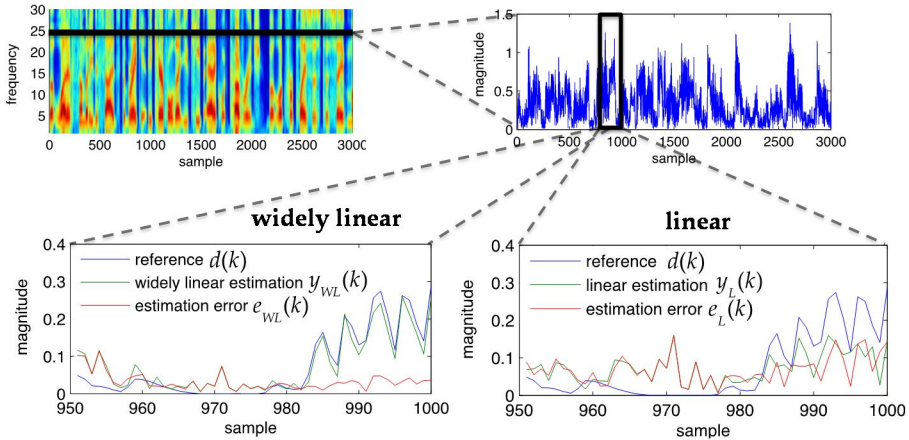


Figure 3.7 - This figure shows the estimation of one source, using a linear extraction vector (lower right plot) and a widely linear one (lower left plot). The upper left plot is the spectrogram of the source. The upper right plot is the time information for some milliseconds, contained in one sub-band of the reference signal. The two lower plots show the amplitude versus time graphic of the local signal separation made by using a widely linear and a linear Wiener extraction vector.

As we have seen, the use of additional information confined in the pseudo-covariance matrix provides important improvements on the algorithms. Following this idea, in the last few years several methods have been developed to beat the performance of classical complex-valued signal processing algorithms in problems like prediction (Javidi, 2008), BSS (Javidi, 2010), and NMF (Looney, 2011), among others.

3.6 The Strong Uncorrelating Transform (SUT)

Following this new point of view, now it is turn to reexamine the blind signal processing section where we present the idea of whitening. It is clear from Theorem 3.3 that the whitening process forces the output to have uncorrelated components, with unit power.

If we apply the whitening matrix \mathbf{W} to general complex observations $\mathbf{x}(k)$, the resulting whitened observations $\mathbf{z}(k)$ have a diagonal covariance matrix. However, what happens with the pseudo-covariance matrix?

Decorrelation does not force any structure on $\tilde{\mathbf{R}}_{z^*}$, so the augmented covariance

matrix \mathbf{R}_z has, in general, non-zero elements outside the main diagonal.

The definition of uncorrelated variables in the complex domain needs to be revised. A complex random vector has *strongly uncorrelated* components if, and only if, its covariance matrix and pseudo-covariance matrix are diagonal. The following theorem from (Koivunen, 2006) states the existence of a matrix that returns uncorrelated components.

Theorem 3.6 (Strong Uncorrelation)

Any complex vector of random processes $\mathbf{x}(k) \in \mathbb{C}^n$ can be transformed by using a nonsingular square matrix $\mathbf{C} \in \mathbb{C}^{n \times n}$, such that the resulting vector of random processes $\mathbf{z}(k) = \mathbf{C}\mathbf{x}(k)$ has the following second-order properties:

$$\mathbf{R}_z = \mathbf{I}, \quad (3.97)$$

and

$$\tilde{\mathbf{R}}_z = \text{diag}(\lambda_1, \dots, \lambda_n), \quad (3.98)$$

where $\lambda_1 \geq \dots \geq \lambda_n$ are the *circularity*⁸ coefficients.

Proof: Applying the definition of the second-order matrices,

$$\begin{aligned} \mathbf{R}_z &= E\{\mathbf{z}(k)\mathbf{z}^H(k)\} = E\{\mathbf{C}\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{C}^H\} \\ &= \mathbf{C}E\{\mathbf{x}(k)\mathbf{x}^H(k)\}\mathbf{C}^H = \mathbf{C}\mathbf{R}_x\mathbf{C}^H, \end{aligned} \quad (3.99)$$

and

$$\begin{aligned} \tilde{\mathbf{R}}_z &= E\{\mathbf{z}(k)\mathbf{z}^T(k)\} = E\{\mathbf{C}\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{C}^T\} \\ &= \mathbf{C}E\{\mathbf{x}(k)\mathbf{x}^T(k)\}\mathbf{C}^T = \mathbf{C}\tilde{\mathbf{R}}_x\mathbf{C}^T. \end{aligned} \quad (3.100)$$

As \mathbf{R}_z is Hermitian and positive definite, and $\tilde{\mathbf{R}}_z$ is symmetric, there exists a matrix \mathbf{C} such that $\mathbf{C}\mathbf{R}_x\mathbf{C}^H = \mathbf{I}$, and $\mathbf{C}\tilde{\mathbf{R}}_x\mathbf{C}^T$ is a diagonal matrix with non-negative entries (see (Horn, 1985 p. 251) for a proof). Note that the diagonal entries can be ordered by simply permuting the rows of \mathbf{C} . ■

The random vectors processes $\mathbf{z}(k)$ satisfying Theorem 3.6 are called strongly

⁸ The term *circularity* in *circularity coefficients* is not accurate because they only characterize second-order circularity, or properness. The name *properness coefficients* would have been more precise.

uncorrelated. Any strongly uncorrelated random vector is white, because of (3.97), but the reverse is not true. In general, the matrix \mathbf{C} is not unique, but the circularity coefficients are. See (Koivunen, 2006) for a proof that justifies the following procedure to find this matrix.

Definition (Strong Uncorrelating Transform): The matrix \mathbf{C} in Theorem 3.6 is called the *Strong Uncorrelating Transform (SUT)*. It can be constructed by using the following method:

1. Find the matrix $\mathbf{D} = \mathbf{R}_{\mathbf{x}}^{-1/2}$, that is, the inverse of the matrix square root⁹ of $\mathbf{R}_{\mathbf{x}}$.
2. Any symmetric matrix $\tilde{\mathbf{R}}_{\mathbf{D}\mathbf{x}}$ has a special form of *Singular Value Decomposition (SVD)*, known as *Takagi's factorization* (see (Horn, 1985)), given by

$$\tilde{\mathbf{R}}_{\mathbf{D}\mathbf{x}} = \mathbf{D}\mathbf{E}\{\mathbf{x}(k)\mathbf{x}^T(k)\}\mathbf{D}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad (3.101)$$

where \mathbf{U} is unitary and $\mathbf{\Lambda}$ is a diagonal matrix with real entries, sorted in increasing order. Note that $\mathbf{R}_{\mathbf{D}\mathbf{x}}$ is the pseudo-covariance matrix of the vector random process $\mathbf{D}\mathbf{x}$.

3. The SUT is given by $\mathbf{C} = \mathbf{U}^H \mathbf{R}_{\mathbf{x}}^{-1/2}$.

An efficient computation of the Takagi Factorization can be found in (Xu, 2005). For practical purposes, a fast MATLAB implementation ready to use is available for download at (Qiao, 2007).

Definition (Circularity spectrum): The vector $\boldsymbol{\lambda} \in \mathbb{R}^n$ which contains the circularity coefficients $\lambda_i, i = 1, \dots, n$ from the diagonal of $\mathbf{\Lambda}$, is called *circularity spectrum*.

Note that a vector random process is proper if, and only if, all its circularity coefficients λ_i are zero.

This procedure called Strong Uncorrelating Transform is the equivalent to whitening, but it diagonalizes not only the covariance matrix, but the pseudo-covariance matrix too. It can be used to decorrelate non-circular sequences, where the standard procedure fails.

3.7 Conclusions

In this chapter, we wanted to give an overview of some successful methods

⁹ The matrix square root $\mathbf{A}^{1/2}$ of a matrix \mathbf{A} , is the square matrix that satisfies $(\mathbf{A}^{1/2})(\mathbf{A}^{1/2}) = \mathbf{A}$. It can be computed from the Eigenvalue Decomposition $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$, as $\mathbf{A}^{1/2} = \mathbf{V}\mathbf{D}^{1/2}\mathbf{V}^{-1}$.

presented in the complex-valued signal processing literature. By using the concepts presented and described in Chapter 2, we studied the following problems:

- The complex version of the MMSE criterion is solved by the corresponding Wiener filter, a supervised method successfully used in engineering. We presented also the stochastic descent algorithm called Complex Least Mean Square, whose convergence was studied.
- We introduced the field of Blind Signal Separation and Extraction, where one has not any training sequence and has to recover the original sources by exploiting their statistics. The problems of PCA and ICA were briefly presented with some of their corresponding solutions.
- When the signals involved are improper, it is a good idea to use the additional information contained in the pseudo-covariance. The framework of augmented statistics and the widely linear models have shown to improve the performance of the classical techniques when the signals are non-circular. Therefore, we presented the Widely Linear solution to the MMSE criterion and the Strong Uncorrelating Transform, which plays the role of decorrelation with improper signals.

These methods have shown its benefits in numerous scenarios during the last two decades. Fortunately, the methodology and the algorithms proposed to find the solutions, establishes a framework for developing novel signal processing tools for problems that are characterized in the complex domain.

4 BOUNDED COMPONENT ANALYSIS

Every man takes the limits of his own field of vision for the limits of the world.

- Arthur Schopenhauer -

The estimation of signals that has been linearly mixed is a problem with many applications in several fields of the science and the technology (Chichocki, 2002). As we have presented in the Chapter 3, there are both blind and supervised criteria that solve the problem of source extraction and separation. These criteria are usually mathematically expressed in the form of the optimization of cost functions, depending on geometrical or statistical parameters of the problem.

4.1 Introduction

In this chapter, we present a blind and geometric technique for the linear decomposition of the observations, known as *Bounded Component Analysis* (BCA). It is based on the hypothesis of the compactness and Cartesian decomposition of the convex support of the component signals. In difference with ICA algorithms, BCA does not need the independence assumption to work, so it can extract successfully sources that are somehow dependent.

The contribution of this chapter can be circumscribed within the group of works in the field of signal processing and communications that exploit some of the geometric properties (Theis, 2003) of the sources in the design of cost functions. Some examples of these properties are the statistical range used in (Pham, 2000) and (Cruces, 2004), the square shape of the convex boundary in (Erdogan, 2006), the constant modulus in (Van der Veen, 1996), the cardinality in (Gamboa, 1997), and the knowledge of the alphabet of the constellation in (Comon, 2004) and (Talwar, 1996).

In this case, we find that the *normalized convex perimeter* of the estimation output is a suitable cost function for the successful extraction of the sources. That is, one can blindly estimate one of the sources by confining a certain combination of the observations into a convex set of minimum perimeter.

Most of the results and the structure of this chapter are based upon the paper (Cruces, 2010) and (Aguilera, 2012). Some sections have been extended with details and others have been reduced to give more coherence to the exposition and to clarify the key results.

The chapter is organized as follows. In Section 4.2, the signal model and some notation from the theory of convex bodies are presented. We will use this notation when deriving several important results of this chapter. In Section 4.3, the main assumptions of BCA are presented. They will be used later in Section 4.4 to guarantee the identifiability of the mixture and the separability of the sources.

Section 4.5 defines the convex perimeter functional, and shows some of its relevant properties. This convex perimeter is used in Section 4.6 to present the minimization criterion in BCA. The cost function derived is shown to be free of local minima for the extraction problem. In Section 4.7, several topics regarding the practical optimization of the cost function are presented, as the gradient used in an extraction algorithm or the computational complexity. Section 4.8 and Section 4.9 present some simple applications for the blind extraction, separation and decomposition from a given set of observations. Finally, Section

4.10 discusses the conclusions to this chapter.

4.2 Signal Model and Notation

Let us consider the classical linear mixture model for complex processes, in a noiseless situation,

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k), \quad (4.1)$$

where

$$\mathbf{s}(k) = \begin{bmatrix} s_1(k) \\ \vdots \\ s_n(k) \end{bmatrix} \in \mathbb{C}^n \quad (4.2)$$

is a random vector drawn from a stationary complex process of source signals with joint probability density function $f_{\mathbf{s}(k)}(\Re(\mathbf{s}(k)), \Im(\mathbf{s}(k)))$, and

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix} \in \mathbb{C}^m \quad (4.3)$$

is the vector of observations. Hereinafter, due to the fact that the processes involved are stationary, we drop the temporal index k to simplify the notation.

The matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$ represents the linear mixture. In this chapter, $m \geq n$ is supposed, which denotes that the mixture is *determined* or *over-determined*. This means that \mathbf{A} is a full column-rank matrix of complex elements, and that the number of observations is greater or equal than the number of sources. In the Chapter 6, we deal with the *under-determined* case, where $m < n$.

We use the symbol $|\cdot|$ to refer to the modulus of a complex variable. When the argument is a vector, it acts component-wise. On the other hand, when the argument is a set, the same notation will be used to refer to its cardinality. This distinction will always be clear from the context.

4.2.1 Blind Signal Separation and Extraction

As we defined in the previous chapter, the problem of recovering one source blindly from the observations is called BSE, and the problem of recovering all the sources is called BSS. We briefly outline the system model and the notation

used in those circumstances.

Blind Source Extraction

To estimate a source from the mixture, we compute the inner product of the observations \mathbf{x} with a non-zero extraction vector $\mathbf{b} \in \mathbb{C}^m$,

$$y = \mathbf{b}^H \mathbf{x} = \mathbf{g}^H \mathbf{s} \in \mathbb{C}, \quad (4.4)$$

where the vector $\mathbf{g} = \mathbf{A}^H \mathbf{b} \in \mathbb{C}^n$ collects the coefficients of the residual mixture¹⁰ of the sources, at the output. This vector can be decomposed into phase and magnitude parts, $\mathbf{g} = |\mathbf{g}| \Lambda_\theta$, where the operators act component-wise. Explicitly,

$$|\mathbf{g}| = \begin{bmatrix} |\mathcal{g}_1| \\ \vdots \\ |\mathcal{g}_n| \end{bmatrix}, \quad (4.5)$$

collects the magnitudes, and

$$\Lambda_\theta = \text{diag} \left(\begin{bmatrix} e^{j\theta_1} & \dots & e^{j\theta_n} \end{bmatrix} \right) = \begin{bmatrix} e^{j\theta_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{j\theta_n} \end{bmatrix} \quad (4.6)$$

is the matrix which collects the contribution of the phases $\theta_1, \dots, \theta_n$.

Two important questions arise here:

- Which one of the n sources is extracted?

The answer to this question will depend on the optimization criterion used, so we address this problem later in the chapter. Without additional constraints or information about the initialization of the algorithm, in blind problems we do not have control of the chosen source. On the other hand, in supervised problems the source is chosen by the reference. The second question is:

- Which is the value of the residual mixture \mathbf{g} when the source is extracted successfully?

Ideally, when the chosen i -th source is extracted, the vector that relates the

¹⁰ This vector \mathbf{g} is also called *residual mixture vector*, or *global transfer vector*, because it represents the transfer function between the input and the output of the whole system.

sources and the output should have zeros on all its elements, except the i -th one. Mathematically,

$$\mathbf{g} = \rho e^{j\theta_i} \mathbf{e}_i, \text{ with } i \in \{1, \dots, n\}, \quad (4.7)$$

where $\rho = \|\mathbf{g}\|_2 > 0$ is the 2-norm of \mathbf{g} , and θ_i is the phase of the i -th coefficient. This is only possible when the system is determined or over-determined, because in other scenarios we do not have enough degrees of freedom to cancel all the residual coefficients but one.

Blind Source Separation

Let us imagine now that we aim to recover all the sources contained in the vector \mathbf{s} from the mixture. As we have said in Chapter 3, there are two approximations to this problem: to recover them jointly, or to recover them sequentially by an extraction (like in BSE) followed by a deflation.

The vector \mathbf{y} is an estimate $\hat{\mathbf{s}}$ of the vector of sources,

$$\mathbf{y} = \mathbf{B}^H \mathbf{x} = \mathbf{G}^H \mathbf{s} \in \mathbb{C}^n, \quad (4.8)$$

where

$$\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] = (\hat{\mathbf{A}}^+)^H \in \mathbb{C}^{m \times n} \quad (4.9)$$

is the Hermitian of the *Moore-Penrose pseudo-inverse* ⁽⁺⁾ (Golub, 1996) of $\hat{\mathbf{A}}$ (an estimate of \mathbf{A}), and $\mathbf{G} = \mathbf{A}^H \mathbf{B} \in \mathbb{C}^{n \times n}$ is the *residual transfer matrix*, whose (i, j) -element accounts for the conjugate of the transfer from the i -th source s_i to the j -th output y_j .

As in the BSE scenario, in a perfect extraction the residual transfer matrix \mathbf{G} should be diagonal, up to permutations and phase/scaling mismatches.

4.2.2 Theory of convex bodies

In the complex domain, certain operations are easier to understand and to develop if we explore the theory of convex bodies. For a formal treatment of this topic, see (Schneider, 1993).

Let us consider the following mapping between the field of complex numbers \mathbb{C} and the space of two-dimensional real vectors \mathbb{R}^2 ,

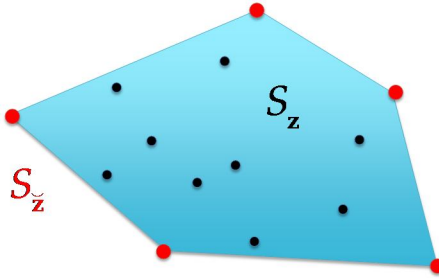


Figure 4.1 - This plot shows a support set S_z defined by all the (red and black) points. The set of red points describes the convex hull \bar{S}_z .

$$z = x + jy = \Re\{z\} + j\Im\{z\} \rightarrow \Gamma(z) = \begin{bmatrix} \Re\{z\} \\ \Im\{z\} \end{bmatrix} \in \mathbb{R}^2. \quad (4.10)$$

This mapping preserves addition and multiplication by a real number. Similarly, the next mapping between the field of complex numbers \mathbb{C} and the subfield of real matrices $\mathbb{R}^{2 \times 2}$,

$$z = x + jy = \Re\{z\} + j\Im\{z\} \rightarrow \Upsilon(z) = \begin{bmatrix} \Re\{z\} & -\Im\{z\} \\ \Im\{z\} & \Re\{z\} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad (4.11)$$

represents multiplication and conjugation of complex variables. If we assume that both mappings (4.10) and (4.11) act component-wise on vectors, the extraction output y can be written as,

$$\begin{aligned} \Gamma(y) &= \Gamma(\mathbf{b}^H \mathbf{x}) = \begin{bmatrix} \Re\{\mathbf{b}^H \mathbf{x}\} \\ \Im\{\mathbf{b}^H \mathbf{x}\} \end{bmatrix} = \\ &= \begin{bmatrix} \Re\{\mathbf{b}\} & \Im\{\mathbf{b}\} \\ -\Im\{\mathbf{b}\} & \Re\{\mathbf{b}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix} = \Upsilon^T(\mathbf{b})\Gamma(\mathbf{x}) \in \mathbb{R}^2. \end{aligned} \quad (4.12)$$

The output can also be expressed using the global transfer vector, like in (4.4), as

$$\Gamma(y) = \Gamma(\mathbf{g}^H \mathbf{s}) = \Upsilon^T(\mathbf{g})\Gamma(\mathbf{s}). \quad (4.13)$$

Next, we define some key concepts from the theory of convex bodies, which will be very useful for the rest of the chapter.

Definition (Support): The *support* of a random vector \mathbf{z} , denoted by S_z , is the set of points of the Euclidean space \mathbb{R}^{2n} for which the probability density function is nonzero. That is,

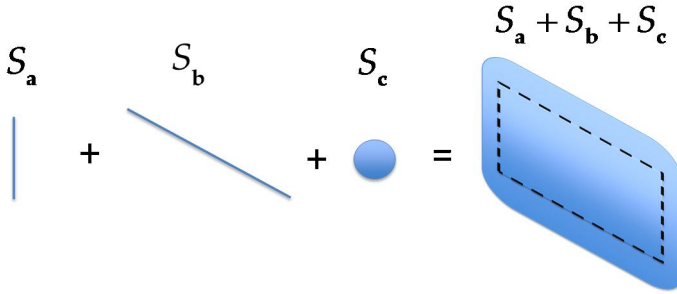


Figure 4.2 - Graphical illustration of the Minkowski sum of two one dimensional sets and one planar set. For completeness, the boundary of the Minkowski sum of the first two one dimensional sets is also represented in dashed line in the final result.

$$S_z = \left\{ \Gamma(\mathbf{z}) \in \mathbb{R}^{2n} : f_z(\Gamma(\mathbf{z})) > 0 \right\}. \quad (4.14)$$

Definition (Convex support): The *convex support* of a random vector \mathbf{z} is the *convex hull* of its support. That is, the smallest convex set that contains S_z . It is denoted by

$$S_z = \text{conv}\{S_z\}. \quad (4.15)$$

The vector $\tilde{\mathbf{z}}$ can be seen as a virtual random vector whose support is the convex body S_z . Hereinafter, we use the notation S_z instead of $S_{\tilde{\mathbf{z}}}$, but it is important to remark that we refer always to convex supports. The Figure 4.1 illustrates the previous concepts.

The main operations to deal with convex sets come from the *Minkowski algebra* (see (Farouki, 2001) for a comprehensive introduction). Consider two random vectors \mathbf{b} and \mathbf{c} , with convex supports S_b and S_c , respectively.

Definition (Minkowski sum): The Minkowski sum of S_b and S_c , is the set

$$S_b + S_c = \left\{ \Gamma(\mathbf{b}) + \Gamma(\mathbf{c}) : \Gamma(\mathbf{b}) \in S_b, \Gamma(\mathbf{c}) \in S_c \right\}, \quad (4.16)$$

which contains all the possible sums of the elements of S_b with the elements of S_c . See Figure 4.2 for an example.

Definition (Minkowski scalar product): For a non-negative real number $k \in \mathbb{R}^+$, the *Minkowski scalar product* kS_c is a dilatation by k of the original set,

$$kS_c = \left\{ k\Gamma(\mathbf{c}) : \Gamma(\mathbf{c}) \in S_c \right\}. \quad (4.17)$$

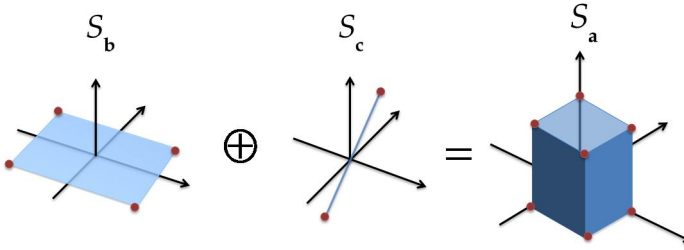


Figure 4.3 - Graphical illustration of the Minkowski direct sum of a planar set and a one dimensional set. The sets lie in complementary linear subspaces, and the result lies in a space of higher dimension.

This scalar product is used later in some proofs. A set growing or shrinking can be mathematical represented as a Minkowski scalar product.

Definition (Minkowski direct sum): When the arguments involved in the Minkowski sum (4.16) lie in complementary linear subspaces with finite dimension, the operation is known as *Minkowski direct sum*, and is represented by $S_b \oplus S_c$. Note that it is important to have the sets lying in complementary subspaces. In Figure 4.3, an example of this kind of sum is shown.

Definition (Directly indecomposable set): A set S_a which cannot be written as a direct sum $S_a = S_b \oplus S_c$, unless for $\dim\{S_b\} = 0$ or $\dim\{S_c\} = 0$, is known as *directly indecomposable*.

The following condition expresses the support of the sum of random vectors as a Minkowski sum of supports.

Theorem 4.1 (Minkowski sum of supports)

A sufficient condition for

$$S_{b+c} = S_b + S_c, \quad (4.18)$$

is given by the following Cartesian product decomposition of the support of the random vector $[\mathbf{b}, \mathbf{c}]$

$$S_{[\mathbf{b}, \mathbf{c}]} = S_b \times S_c, \quad (4.19)$$

where $[\mathbf{b}, \mathbf{c}]$ is the concatenation of the two random vectors \mathbf{b} and \mathbf{c} .

Proof: Using the definition (4.14),

$$\begin{aligned}
S_{\mathbf{b}+\mathbf{c}} &= \left\{ \Gamma(\mathbf{b}+\mathbf{c}) : f_{\mathbf{b}+\mathbf{c}}(\Gamma(\mathbf{b}+\mathbf{c})) > 0 \right\} \\
&= \left\{ \Gamma(\mathbf{b}+\mathbf{c}) : \Gamma([\mathbf{b}, \mathbf{c}]) \in S_{[\mathbf{b}, \mathbf{c}]} \right\} \\
&= \left\{ \Gamma(\mathbf{b}) + \Gamma(\mathbf{c}) : \Gamma([\mathbf{b}, \mathbf{c}]) \in S_{[\mathbf{b}, \mathbf{c}]} \right\}.
\end{aligned} \tag{4.20}$$

By using the Cartesian product decomposition (4.19),

$$S_{[\mathbf{b}, \mathbf{c}]} = S_{\mathbf{b}} \times S_{\mathbf{c}} = \left\{ \Gamma(\mathbf{b}) \in S_{\mathbf{b}}, \Gamma(\mathbf{c}) \in S_{\mathbf{c}} \right\}, \tag{4.21}$$

so

$$S_{\mathbf{b}+\mathbf{c}} = \left\{ \Gamma(\mathbf{b}) + \Gamma(\mathbf{c}) : \Gamma(\mathbf{b}) \in S_{\mathbf{b}}, \Gamma(\mathbf{c}) \in S_{\mathbf{c}} \right\} = S_{\mathbf{b}} + S_{\mathbf{c}}. \tag{4.22}$$

■

For some given support sets $S_{\mathbf{b}}$ and $S_{\mathbf{c}}$, Theorem 4.1 establishes the condition that guarantees that $S_{[\mathbf{b}, \mathbf{c}]}$ attains the maximum possible volume, relative to the smallest subspace in which this set lies. Of course, condition (4.19) is satisfied when the random vectors are mutually independent. However, the converse is not true, so the random vectors can be mutually dependent and the condition holds true.

4.3 Main Assumptions of Bounded Component Analysis

As the mutual independence condition in ICA, or the joint decorrelation of the sources in PCA, to perform a latent component analysis it is necessary to establish some assumptions.

For the Bounded Component Analysis of the observations, we need to assume the following properties for the sources and the mixture:

1. *Compactness and non-degeneracy of the sources:* we assume that all sources are non-degenerate random variables of compact support. This means that the support of its probability density function is not only a single point, and that its support is finite.
2. *Cartesian decomposition of the convex support of the sources:* the minimum convex support of the random vector of sources \mathbf{s} , can be decomposed as the Cartesian product (4.19) of the individual convex support of the sources,

$$S_{\mathbf{s}} = S_{s_1} \times \cdots \times S_{s_n}. \tag{4.23}$$

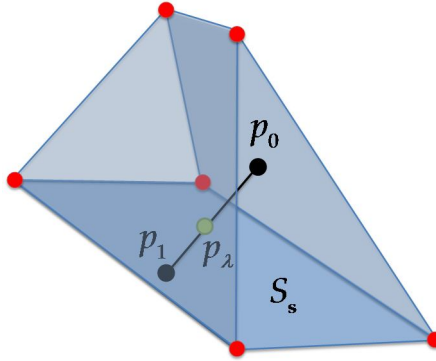


Figure 4.4 - This is a 3-dimensional representation of a convex set. Suppose that all the blue space is filled with points. The extreme points are shown in red. The number of extreme points on a convex set is relatively small, when compared with the total number of points on the set.

3. *Lossless mixing*: the mixing system is characterized by a full column-rank mixing matrix $\mathbf{A} \in \mathbb{C}^{m \times n}$, with $m \geq n$.

Property 3 guarantees that the mixing system is invertible, thus allowing the perfect recovery of the sources. Properties 1 and 2 replace the hypothesis of the mutual independence of the sources in ICA, which is no longer necessary. Although it can be seen as a very strict constraint, this is not the case. That is because Property 2 relies only on the *extreme points* of the convex set.

Definition (Extreme points): Extreme points of a convex set S_s , denoted by $\text{ext}\{S_s\}$, are those points of the set which cannot be expressed as a convex combination

$$p_\lambda = (1 - \lambda)p_0 + \lambda p_1, \quad (4.24)$$

where p_0, p_1 are two different arbitrary points of S_s , with $0 < \lambda < 1$. In Figure 4.4, the extreme points of a 3-dimensional convex set are shown.

A convex and compact set is the closed convex hull of its extreme points (see (Rockafellar, 1970 p. 167) for a proof). Thus, once the compactness of the support set of the sources in Property 1 is assumed, Property 2 is equivalent to the following Cartesian decomposition of the set of extreme points,

$$\text{ext}\{S_s\} = \text{ext}\{S_{s_1}\} \times \cdots \times \text{ext}\{S_{s_n}\}. \quad (4.25)$$

The hardness of this hypothesis depends on the cardinality of (4.25), that is,

$$\left| \text{ext} \{ S_s \} \right| = \prod_{i=1}^n \left| \text{ext} \{ S_{s_i} \} \right|. \quad (4.26)$$

which, for identically distributed sources, increases polynomially with the number of extreme points $\text{ext} \{ S_s \}$, and exponentially with the number of sources n . For example, the typical scenario in communications is characterized by few extreme points, and few sources of interference, so the hypothesis can be regarded as weak.

The following theorem states the existence of a direct decomposition of the convex hull of the observations \mathbf{x} , in a complex linear mixture model.

Theorem 4.2 (Direct decomposition of the convex hull of the observations)

Suppose the following linear model for the observations,

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{i=1}^n \mathbf{a}_i s_i, \quad (4.27)$$

where \mathbf{a}_i is the i -th column of the mixture matrix \mathbf{A} . When the previous assumptions 1-3 hold true, there exist a decomposition of the convex support of the observations, S_x , as a Minkowski direct sum of n convex summands. That is,

$$S_x = S_{\mathbf{a}_1 s_1} \oplus \cdots \oplus S_{\mathbf{a}_n s_n}. \quad (4.28)$$

Proof: Form (4.23) in Property 2, we can see that the convex support of the vector

$$\begin{bmatrix} \mathbf{a}_1 s_1 \\ \vdots \\ \mathbf{a}_n s_n \end{bmatrix}, \quad (4.29)$$

which is obtained by stacking all the component vectors of the observations in one column, decomposes as a Cartesian product of the convex supports of the component vectors. Mathematically, this is a concatenation like the one used in (4.19), so

$$S_{\begin{bmatrix} \mathbf{a}_1^T s_1, \dots, \mathbf{a}_n^T s_n \end{bmatrix}^T} = S_{\mathbf{a}_1 s_1} \times \cdots \times S_{\mathbf{a}_n s_n}. \quad (4.30)$$

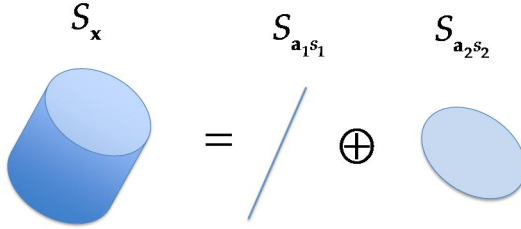


Figure 4.5 - This figure illustrates, for a 2×2 invertible mixture of two sources, the decompositions of the support of the observations as a Minkowski direct sum of two sets: $S_{\mathbf{a}_1 s_1}$ and $S_{\mathbf{a}_2 s_2}$. Since $m = 2$, the convex support of the observations is a set embedded in the four-dimensional real space. Hence, we have used a projection to represent it in only three dimensions. This projection was chosen so as to preserve invariant $S_{\mathbf{a}_2 s_2}$ while reducing the planar $S_{\mathbf{a}_1 s_1}$ set to a single dimension, as it is shown in the figure.

And thus, we are in the conditions of Theorem 4.1. After applying it, we obtain

$$S_{\mathbf{x}} = S_{\sum_{i=1}^n \mathbf{a}_i s_i} = S_{\mathbf{a}_1 s_1} + \dots + S_{\mathbf{a}_n s_n}. \quad (4.31)$$

Finally, by assumption 3, the linear independence of all the column vectors $\mathbf{a}_1, \dots, \mathbf{a}_n$ of the mixing matrix makes the Minkowski sum (+) equal to the Minkowski direct sum (\oplus) in equation (4.28). ■

This theorem is key to develop our proposed Bounded Component Analysis of the observations. It states how to decompose the convex support of the observations into bounded components. In Figure 4.5, an example of this decomposition of the observations into components is shown.

4.4 Identifiability, Separability, and Indeterminacies

In this section we aim to answer an important question which arises in signal separation. For a sufficiently long sequence of observations \mathbf{x} , does our model uniquely determine the columns of the mixing system \mathbf{a}_i and the sources \mathbf{s} ?

This problem has been extensively studied in the last years. For example, when the sources shares a known and finite alphabet, identifiability results based on algebraic and geometric arguments can be derived, as shown in (Talwar, 1996) or (Comon, 2004).

In ICA, when the mixing system is full-column rank, and there are no two Gaussian sources in the mixture with the same circularity coefficient, the extension of the Darmois-Skitovitch theorem (Koivunen, 2006) guarantees the identifiability of the mixing system \mathbf{A} . These conditions guarantee the separability of the sources too, up to a permutation, scaling and a phase term.

In BCA, we must determine whether the decomposition of S_x , presented in Theorem 4.2, is unique or not. For a set of n' linearly independent vectors $\mathbf{a}_i', \dots, \mathbf{a}_{n'}'$, we aim to find a geometric decomposition of the observations vector

$$\mathbf{x} = \sum_{i=1}^{n'} \mathbf{a}_i' s_i' , \quad (4.32)$$

different from (4.27), such that

$$S_x = S_{\mathbf{a}_1' s_1'} \oplus \dots \oplus S_{\mathbf{a}_{n'}' s_{n'}'} . \quad (4.33)$$

The following theorem provides the answer to this question.

Theorem 4.3 (Essential uniqueness of the decomposition)

The decomposition of S_x , given in Theorem 4.2, is essentially unique, up to possible permutation of the direct summands.

We refer the reader to Appendix A in (Cruces, 2010) for a proof. It is based on the essential uniqueness of the direct decomposition of convex bodies in terms of directly decomposable bodies (see (Schneider, 1993)).

According to Theorem 4.3, there exists a permutation π of $1, \dots, n$ such that $S_{\mathbf{a}_i s_i} = S_{\mathbf{a}_{\pi_i} s_{\pi_i}}$, for $i = 1, \dots, n$. By identifying each one of these sets, we are uniquely determining a column of the mixing system up to permutation, scaling, and phase ambiguities (see equation (4.7)). These are the same kind of indeterminacies of the ICA model.

The separability of the sources is guaranteed due to the full column-rank assumption (Property 3 of BCA), up to the same well-known indeterminacies.

When extracting one source, the extraction vector \mathbf{b}^H is proportional to one of the rows of the Moore-Penrose pseudo-inverse of \mathbf{A} , denoted as \mathbf{A}^+ . Mathematically, these vectors \mathbf{b} belongs to the set E_b , with

$$E_b = \left\{ \rho e^{j\theta_i} \mathbf{a}_i^\# : \rho > 0, \theta_i \in [0, 2\pi), i = 1, \dots, n \right\}, \quad (4.34)$$

where $\mathbf{a}_i^\#$ is the i -th column of $(\mathbf{A}^+)^H$, i.e.,

$$\mathbf{a}_i^\# = \left(\mathbf{A}^+\right)^H \mathbf{e}_i. \quad (4.35)$$

The scaling indeterminacy can be removed by assuming that the sources have unit variance, so hereinafter we make this hypothesis.

At this point, we need a metric to quantify whether the sources are bounded or not. The next section presents a scalar functional with some nice algebraic properties.

4.5 The Convex Perimeter of the Output

This section defines the convex perimeter of the output of a linear system. We present also some of its relevant properties, which are crucial to design an algorithm that rewards whether the signals are bounded.

From equation (4.4), the output y estimates one of the sources and is the coordinate of the orthogonal projection of the observations onto the span of the extraction vector \mathbf{b} . Due to Theorem 4.1, the support of the output S_y can be decomposed as

$$S_y = S_{\sum_{i=1}^n g_i^* s_i} = \sum_{i=1}^n S_{g_i^* s_i} = \sum_{i=1}^n |g_i| S_{e^{-j\theta_i} s_i}. \quad (4.36)$$

Definition (Convex perimeter): The *convex perimeter* of a random variable is the perimeter of the convex hull of the support set of the random variable.

Let $V_2(\cdot)$ denote the Lebesgue measure (Croft, 1991 p. 4) in \mathbb{R}^2 (the area), and let S_B be the ball of unit radius. The convex perimeter of y is defined as the growth rate of the area functional of the outer parallel body of S_y , at an arbitrary small distance ε :

$$L(y) \equiv \ell(S_y) = \lim_{\varepsilon \rightarrow 0^+} \frac{V_2(S_y + \varepsilon S_B) - V_2(S_y)}{\varepsilon} \in \mathbb{R}^+. \quad (4.37)$$

This definition is graphically illustrated in Figure 4.6. As the ring between the outer and the inner convex set gets thinner, the difference between the areas, divided by the distance between them, becomes the perimeter.

The following properties are very useful to deal with this real scalar value.

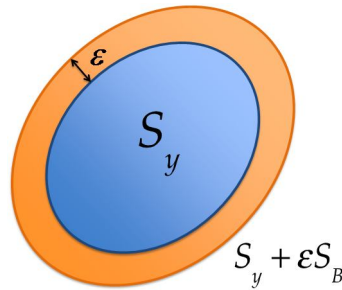


Figure 4.6 - The definition of the convex perimeter (4.37) is illustrated here. The value ε is the distance between the outer and the inner set. The outer set is auxiliary and only exists to help in the definition of $\ell(S_y)$.

Theorem 4.4 (Properties of the convex perimeter)

The convex perimeter is a functional $\ell \in \mathbb{R}$, and it has the following properties:

- 1) Continuity.
- 2) Minkowski additivity.

$$\ell(S_1 + S_2) = \ell(S_1) + \ell(S_2) \quad (4.38)$$

- 3) Positive homogeneity of degree one.

$$\ell(\alpha S_1) = \alpha \ell(S_1) \quad , \quad \alpha \in \mathbb{R}^+ \quad (4.39)$$

- 4) Invariance under rigid motions (translations/rotations) of its argument.

Proof: Consider two convex domains S_1 and S_2 . The area of the convex domain $S = \alpha_1 S_1 + \alpha_2 S_2$, with $\alpha_1, \alpha_2 > 0$, can be expressed as a quadratic polynomial in α_1, α_2 , given by

$$V_2(\alpha_1 S_1 + \alpha_2 S_2) = \sum_{i_1, i_2}^2 \alpha_{i_1} \alpha_{i_2} V(S_{i_1}, S_{i_2}) \quad (4.40)$$

The real coefficient $V(S_1, S_2)$ is called the *mixed volume* of S_1, S_2 , and it is a fundamental concept in the theory of convex bodies (see (Weil, 1975)). Also, it satisfies the following properties:

$$a) \quad V_2(S) = V(S, S), \text{ for all convex set } S.$$

b) The mixed volume is additive in each variable, that is

$$V(S_1 + S_1', S_2) = V(S_1, S_2) + V(S_1', S_2) \quad , \quad \forall S_1, S_1', S_2. \quad (4.41)$$

c) The mixed volume is symmetric in all variables.

When we apply these properties a) - c) in the definition of the perimeter (4.37), it results twice the mixed volume of S_y and the ball of unit radius S_B ,

$$\begin{aligned} \ell(S_y) &= \lim_{\varepsilon \rightarrow 0^+} \frac{V_2(S_y + \varepsilon S_B) - V_2(S_y)}{\varepsilon} \\ &= \lim_{\varepsilon \rightarrow 0^+} \frac{V(S_y, S_y) + \varepsilon^2 V(S_B, S_B) + 2\varepsilon V(S_y, S_B) - V_2(S_y)}{\varepsilon} \\ &= 2V(S_y, S_B). \end{aligned} \quad (4.42)$$

Now, if we have two convex sets S_1 and S_2 , then the perimeter of their sum is given by property b),

$$\begin{aligned} \ell(S_1 + S_2) &= 2V(S_1 + S_2, S_B) \\ &= 2V(S_1, S_B) + 2V(S_2, S_B) \\ &= \ell(S_1) + \ell(S_2). \end{aligned} \quad (4.43)$$

And that proves Property 2 in Theorem 4.4.

To prove Property 3, suppose that $\alpha \in \mathbb{R}^+$ is a positive real constant. Using again the definition of the mixed volume (4.40), and noting that the area satisfies $V_2(\alpha S_y) = \alpha^2 V_2(S_y)$, the convex perimeter of the scaled set is

$$\begin{aligned} \ell(\alpha S_y) &= \lim_{\varepsilon \rightarrow 0^+} \frac{V_2(\alpha S_y + \varepsilon S_B) - V_2(\alpha S_y)}{\varepsilon} \\ &= \lim_{\varepsilon \rightarrow 0^+} \frac{\alpha^2 V(S_y, S_y) + \varepsilon^2 V(S_B, S_B) + 2\varepsilon \alpha V(S_y, S_B) - \alpha^2 V_2(S_y)}{\varepsilon} \\ &= 2\alpha V(S_y, S_B) = \alpha \ell(S_y). \end{aligned} \quad (4.44)$$

Moreover, if we use the definition of the mixed volume (4.40) to expand the area of the sum of S and the ball of unit radius,

$$\begin{aligned}
V_2(S_y + S_B) &= V(S_y, S_B) + V(S_y, S_y) + V(S_y, S_B) + V(S_B, S_B) \\
&= 2V(S_y, S_B) + V_2(S_y) + V_2(S_B) \\
&= \ell(S_y) + V_2(S_y) + V_2(S_B),
\end{aligned} \tag{4.45}$$

we can write the perimeter in terms of area measures,

$$\ell(S_y) = V_2(S_y + S_B) - V_2(S_y) - V_2(S_{yB}). \tag{4.46}$$

Finally, as the area functional $V_2(\cdot)$ of a convex body is continuous and invariant to rigid motions, the perimeter functional inherits these properties from (4.46), resulting in properties 1 and 4 in Theorem 4.4 and finishing this proof. ■

The next result uses the previous properties of the convex perimeter to understand it as an inner product of two vectors. Let us assume that the perimeter acts component-wise on random vectors, i.e.,

$$L(\mathbf{s}) = \mathbf{L}_S = \begin{bmatrix} L(s_1) \\ \vdots \\ L(s_n) \end{bmatrix} \in \mathbb{R}^n. \tag{4.47}$$

Theorem 4.5 (Inner product interpretation)

The convex perimeter of the output $L(y)$ provides an implicit evaluation of the inner product between the vectors $|\mathbf{g}|$ and \mathbf{L}_S . That is,

$$L(y) = \langle |\mathbf{g}|, \mathbf{L}_S \rangle = \sum_{i=1}^n |g_i| L(s_i), \tag{4.48}$$

where

$$\mathbf{L}_S = \begin{bmatrix} L(s_1) \\ \vdots \\ L(s_n) \end{bmatrix} \in \mathbb{R}^n. \tag{4.49}$$

Proof: If we attend at the decomposition (4.36) of the convex set S_y , we can apply, in this order, Property 2 (additivity) and Property 3 (homogeneity) of Theorem 4.4,

$$\begin{aligned}
L(y) &\equiv \ell(S_y) = \ell\left(\sum_{i=1}^n |g_i| S_{e^{-j\theta_i} s_i}\right) \\
&= \sum_{i=1}^n \ell(|g_i| S_{e^{-j\theta_i} s_i}) = \sum_{i=1}^n |g_i| \ell(S_{e^{-j\theta_i} s_i})
\end{aligned} \tag{4.50}$$

And now, applying Property 4 (invariance to rotations),

$$L(y) = \sum_{i=1}^n |g_i| \ell(S_{s_i}) = \sum_{i=1}^n |g_i| L(s_i), \tag{4.51}$$

resulting on the inner product of $|\mathbf{g}|$ and \mathbf{L}_S .

■

Once we have exposed the properties of the convex perimeter, we are able to present the optimization criterion in which BCA is based.

4.6 The Criterion of Minimum Normalized Convex Perimeter

At this point, we are ready to introduce the criterion of minimum normalized convex perimeter of the output. Consider the function $\psi(\mathbf{b}) : D_{\mathbf{b}} \rightarrow \mathbb{R}^+$, where

$$D_{\mathbf{b}} = \{\mathbf{b} \in \mathbb{C}^m \setminus \{\mathbf{0}\}\} \tag{4.52}$$

is the set of all vectors in \mathbb{C}^m but the origin. This cost function takes only positive values, and is defined by

$$\psi(\mathbf{b}) = \frac{L(y)}{\|\mathbf{b}\|} = \frac{L(\mathbf{b}^H \mathbf{x})}{\|\mathbf{b}\|}. \tag{4.53}$$

That is, our cost function is the convex perimeter of the output, normalized by the 2-norm of the extraction vector. The *criterion of minimum normalized convex perimeter* aims to minimize, using a stochastic descent algorithm, the cost function (4.53),

$$\mathbf{b}_{opt} = \arg \min_{\mathbf{b}} \{\psi(\mathbf{b})\} = \arg \min_{\mathbf{b}} \left\{ \frac{L(\mathbf{b}^H \mathbf{x})}{\|\mathbf{b}\|} \right\}. \tag{4.54}$$

The normalization is useful to force the extraction vector to be non-zero. If we do not normalize by $\|\mathbf{b}\|$, a minimization algorithm could select the optimal

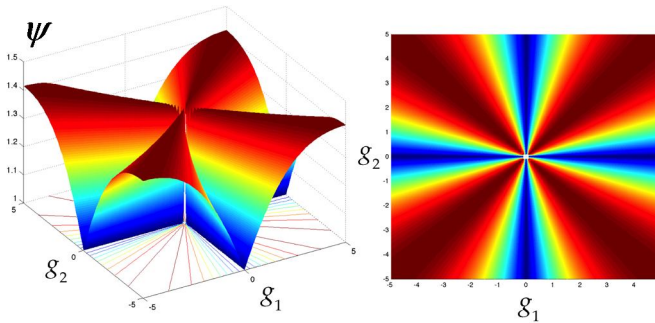


Figure 4.7 - This is a representation of the value of the cost function (4.53) versus the $m=2$ coefficients of the residual transfer vector. The left figure shows a 3D viewpoint, while the right figure shows a central viewpoint. For simplicity, all the values involved are real. Red zones correspond to high values, and blue zones correspond to low values.

solution as $\mathbf{b} = \mathbf{0}$, because indeed it results in an output with zero perimeter $y = 0$. Figure 4.7 shows the shape of the cost function. Note that there are several subspaces corresponding to solutions of the criterion (the dark blue zones). As we will see in the next section, each one of the solutions corresponds to the extraction of a source, up to scaling and phase ambiguities. Also, it can be seen that there is a singularity at the origin.

Let us look further into the properties of the cost function (4.53).

4.6.1 Properties of the normalized convex perimeter

The properties of homogeneity and rigid motion invariance of the perimeter functional (3 and 4 in Theorem 4.4), imply that the cost function is invariant under complex multiplications of the argument,

$$\psi(c\mathbf{b}) = \psi(\mathbf{b}) \quad , \quad \forall c \in \mathbb{C} \setminus \{0\}. \quad (4.55)$$

Consider the family of subsets $P_{\mathbf{b}}(\rho, \boldsymbol{\theta})$ of the domain of extraction vectors $D_{\mathbf{b}}$ (see (4.52)), for which the residual mixing vectors have constant 1-norm¹¹ of value ρ and constant phase $\boldsymbol{\theta} = [\theta_1 \quad \cdots \quad \theta_n]^T$. That is,

$$P_{\mathbf{b}}(\rho, \boldsymbol{\theta}) = \left\{ \mathbf{b} = (A^+)^H \mathbf{g} : \|\mathbf{g}\|_1 = \rho \right\}. \quad (4.56)$$

¹¹ The 1-norm of a vector is a real-valued scalar defined as $\|\mathbf{g}\|_1 = \sum_{i=1}^n |g_i| \in \mathbb{R}$. Note that it is important to put the sub-index "1" in the definition of the norm $\|\cdot\|_1$.

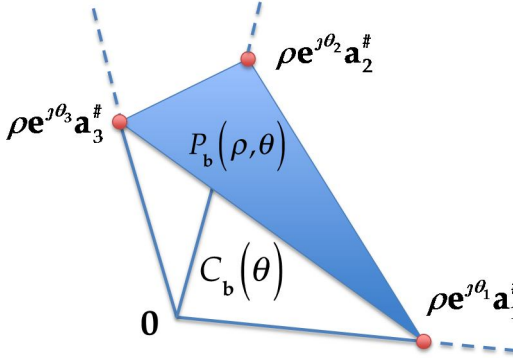


Figure 4.8 - View of one of the polytopes $P_b(\rho, \theta)$ within the convex cone $C_b(\theta)$. The vertices of the polytope play a key role in the analysis of the solutions of the cost function.

Each of these subsets is compact, convex, and can be represented as a finite intersection of half-spaces. Therefore, it is also a *polytope*. The infinite union of these polytopes, for a constant θ and all the possible values of $\rho > 0$, defines the *convex cone* (see the complete reference (Boyd, 2009)) of extraction vectors

$$C_b(\theta) = \bigcup_{\forall \rho > 0} P_b(\rho, \theta). \quad (4.57)$$

The figure Figure 4.8 shows this convex cone as an intersection of polytopes. The following theorem is the first step to analyze the geometry and the solutions of the cost function.

Theorem 4.6 (Quasiconcavity)

Within each convex cone $C_b(\theta)$, the function $\psi(\mathbf{b})$ is strictly *quasiconcave*. That is, for any $\mathbf{b}_0, \mathbf{b}_1 \in C_b(\theta)$ and $\lambda \in (0, 1)$, we have that

$$\psi((1-\lambda)\mathbf{b}_0 + \lambda\mathbf{b}_1) > \min\{\psi(\mathbf{b}_0), \psi(\mathbf{b}_1)\} \quad (4.58)$$

Proof: Due to the convexity of the convex cone $C_b(\theta)$, for any $\mathbf{b}_0, \mathbf{b}_1 \in C_b(\theta)$ and $\lambda \in [0, 1]$, we have that

$$\mathbf{b}_\lambda = (1-\lambda)\mathbf{b}_0 + \lambda\mathbf{b}_1 \in C_b(\theta). \quad (4.59)$$

Let us denote by $\mathbf{g}_0, \mathbf{g}_1$ and \mathbf{g}_λ the residual mixing vectors which, under the transformation $\mathbf{g} = \mathbf{A}^H \mathbf{b}$, correspond to $\mathbf{b}_0, \mathbf{b}_1$ and \mathbf{b}_λ , respectively. Using (4.51) the convex perimeter $L(\mathbf{b}_\lambda^H \mathbf{x})$ can be expanded as

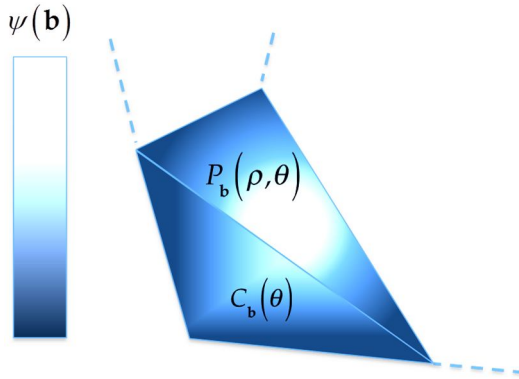


Figure 4.9 – This figure shows conceptually the value of the cost function $\psi(\mathbf{b})$ on the polytope $P_b(\rho, \theta)$, and in part of the frontier of the convex cone $C_b(\theta)$.

$$L(\mathbf{b}_\lambda^H \mathbf{x}) = \sum_{i=1}^n |(g_\lambda)_i| L(s_i). \quad (4.60)$$

Applying the relation (4.59) to (4.60), but using \mathbf{g} instead of \mathbf{b} ,

$$\begin{aligned} L(\mathbf{b}_\lambda^H \mathbf{x}) &= \sum_{i=1}^n \left((1-\lambda) |(g_0)_i| + \lambda |(g_1)_i| \right) L(s_i) \\ &= (1-\lambda) L(\mathbf{b}_0^H \mathbf{x}) + \lambda L(\mathbf{b}_1^H \mathbf{x}) \end{aligned} \quad (4.61)$$

On the other hand, if we apply the triangular inequality (see, for instance, (Apostol, 1967 p. 42)) to the convexity condition (4.59), it results in

$$\|\mathbf{b}_\lambda\| < (1-\lambda)\|\mathbf{b}_0\| + \lambda\|\mathbf{b}_1\|, \quad \lambda \in (0, 1). \quad (4.62)$$

If we divide (4.61) by (4.62), we obtain the cost function

$$\psi_x(\mathbf{b}_\lambda) = \frac{L(\mathbf{b}_\lambda^H \mathbf{x})}{\|\mathbf{b}_\lambda\|} > \frac{(1-\lambda)L(\mathbf{b}_0^H \mathbf{x}) + \lambda L(\mathbf{b}_1^H \mathbf{x})}{(1-\lambda)\|\mathbf{b}_0\| + \lambda\|\mathbf{b}_1\|}. \quad (4.63)$$

Finally, we apply the useful inequality

$$\frac{a+b}{c+d} \geq \min \left\{ \frac{a}{c}, \frac{b}{d} \right\}, \quad \text{with } a, b, c, d \in \mathbb{R}. \quad (4.64)$$

Thus, the relation (4.63) results in

$$\psi(\mathbf{b}_\lambda) > \min \left\{ \frac{L(\mathbf{b}_0^H \mathbf{x})}{\|\mathbf{b}_0\|}, \frac{L(\mathbf{b}_1^H \mathbf{x})}{\|\mathbf{b}_1\|} \right\} = \min \{ \psi(\mathbf{b}_0), \psi(\mathbf{b}_1) \}. \quad (4.65)$$

So the quasiconcavity of the cost function is shown. ■

The Theorem 4.6 is illustrated in Figure 4.9. The quasiconcavity property helps us to show that the set of extraction vectors, which are the valid solutions of the problem, contains all the minima of the contrast function.

The next theorem states that the cost function has no spurious extreme points, in absence of noise. This is a very desirable property, because it ensures that a gradient descent algorithm, with the proper step size and initialization point, may reach any optimal solution.

Theorem 4.7 (Global cost function)

The normalized convex perimeter of the output, $\psi(\mathbf{b})$, is a scale and phase invariant contrast function, whose local and global minima always correspond to valid solutions of the extraction problem.

Proof: As we have seen in Theorem 4.6, for each of the convex cones (4.57), the cost function proposed in (4.53) is quasiconcave.

From the convexity of the polytope $P_b(\rho, \theta)$, any convex combination $\mathbf{b}_\lambda = (1-\lambda)\mathbf{b}_0 + \lambda\mathbf{b}_1$ of two different arbitrary points $\mathbf{b}_0, \mathbf{b}_1 \in P_b(\rho, \theta)$, with $\lambda \in (0, 1)$, also belongs to $P_b(\rho, \theta)$.

However, \mathbf{b}_λ cannot be a minimum of the function relative to $P_b(\rho, \theta)$, because the continuity of the function and Theorem 4.6 guarantees the existence of a descent direction.

Therefore, the minima should necessarily belong to the extreme points of the set. However, the extreme points of the compact and convex polytope $P_b(\rho, \theta)$ are its vertices. As we can see in the Figure 4.8, these vertices are located in the direction of the columns of the matrix \mathbf{A}^+ ,

$$\text{ext}\{P_b(\rho, \theta)\} = \text{vert}\{P_b(\rho, \theta)\} = \{\rho e^{j\theta_1} \mathbf{a}_1^\#, \dots, \rho e^{j\theta_n} \mathbf{a}_n^\#\}. \quad (4.66)$$

If we apply this argument to each of the possible polytopes $P_b(\rho, \theta)$, and unite them all into the whole domain of the function,

$$D_{\mathbf{b}} = \bigcup_{\forall \boldsymbol{\theta}, \forall \rho > 0} P_{\mathbf{b}}(\rho, \boldsymbol{\theta}), \quad (4.67)$$

we obtain the set that contains all the minima \mathbf{b} of the cost function (4.53),

$$\begin{aligned} E_{\mathbf{b}} &= \text{ext}\{D_{\mathbf{b}}\} = \bigcup_{\forall \boldsymbol{\theta}, \forall \rho > 0} \text{ext}\{P_{\mathbf{b}}(\rho, \boldsymbol{\theta})\} \\ &= \left\{ \rho e^{j\theta_i} \mathbf{a}_i^{\#} : \rho > 0, \theta_i \in [0, 2\pi), i = 1, \dots, n \right\}. \end{aligned} \quad (4.68)$$

This set is exactly the same as in (4.34), that is, the whole set of vectors \mathbf{b} that extract one of the sources. Thus, all the minima of the function, either local or global, constitute a valid solution to the extraction problem. ■

This property of absence of wrong solutions is also shared by a few extraction criteria based on high-order statistics or geometry. At this point, we aim to answer an interesting question: what happens when some of the sources are real-valued? Is the minimum normalized convex perimeter criterion consistent?

4.6.2 Consistency of the real case

The criterion of minimum normalized convex perimeter,

$$\mathbf{b}_{opt} = \arg \min_{\mathbf{b}} \left\{ \frac{L(\mathbf{b}^H \mathbf{x})}{\|\mathbf{b}\|} \right\}, \quad (4.69)$$

can work with arbitrary complex-valued sources, as long as they satisfy the BCA assumptions 1-3 established in Section 4.3. Hence, this includes as special cases, those where some or all the sources are real valued. In these scenarios, it is important to remark that the convex perimeter of a real random variable is twice their *statistical range*¹²,

$$\mathbf{b}_{opt} = \arg \min_{\mathbf{b}} \left\{ \frac{2(\max\{y\} - \min\{y\})}{\|\mathbf{b}\|} \right\}, \quad y \in \mathbb{R}. \quad (4.70)$$

There exists the extreme case where the mixture matrix and all the sources are real. In this case, the criterion proposed in (4.69) particularizes to the minimization of the normalized statistical range of the output, as presented in

¹² The statistical range of a real random variable is the maximum of the distance between any two points of the support of its probability density function.

the works (Pham, 2000), (Cruces, 2004), and (Vrins, 2005). To avoid numerical problems, in practical implementations one may add a very small imaginary part to the sources.

4.7 Optimization of the Cost Function

In this section, we present some of the most important issues that need to be solved if we want to construct a useful algorithm that works well, fast and reliable.

The whitening procedure is a common practice before doing an extraction or separation from a set of observations. However, the proposed criterion of minimum normalized convex perimeter does not need this assumption, as happens in other approaches.

In practice, that preprocessing step is recommended because it helps to achieve the convergence of the methods to a successful solution. Also, in the next chapter, whitening the observations helps us to analyze the convergence behavior of a BCA extraction algorithm.

4.7.1 Whitening as a pre-processing step

Whitening, as we have seen in Section 3.4.1, consists on finding a matrix $\mathbf{W} \in \mathbb{C}^{n \times m}$ such as

$$\mathbf{z} = \mathbf{W}\mathbf{x}, \quad (4.71)$$

where \mathbf{z} is a whitened complex random vector whose correlation matrix is the identity. That means that the variables within are uncorrelated and with unit variance. See Theorem 3.3 for more information.

However, if it is necessary, the matrix \mathbf{W} can be modified to reduce the dimension of the observation set, supposed that some of the signals are somehow dependent.

The *regular mixing matrix*, \mathbf{V} , combines the effects of whitening and mixing,

$$\mathbf{V} = \mathbf{W}\mathbf{A} \in \mathbb{C}^{n \times n}. \quad (4.72)$$

The whitened observations \mathbf{z} are directly related to the sources by the linear mixing model

$$\mathbf{z} = \mathbf{V}\mathbf{s} \in \mathbb{C}^n. \quad (4.73)$$

Note how \mathbf{V} acts now as the *unitary mixing matrix*. There exists an *extraction*

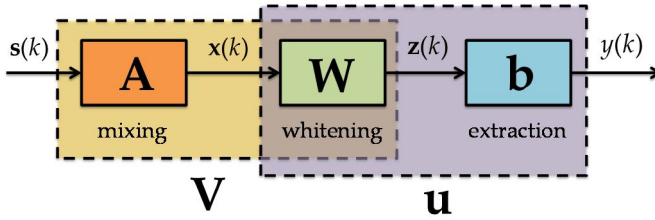


Figure 4.10 - Block diagram of the mixing, whitening, and extraction systems.

vector \mathbf{u} that gives the output

$$y = \mathbf{u}^H \mathbf{z} = \mathbf{u}^H \mathbf{V} \mathbf{s} = \mathbf{g}^H \mathbf{s} \in \mathbb{C}, \quad (4.74)$$

where the residual mixing vector is related to the extraction vector like $\mathbf{g} = \mathbf{V}^H \mathbf{u}$. The vector \mathbf{u} plays the role of the extraction vector \mathbf{b} in the simplest case where they are not whitened.

In Figure 4.10, the system including the whitening matrix is shown to clarify the notation presented in this subsection. We now propose a simple, although adequate, gradient descent method to optimize the cost function (4.53).

4.7.2 Differentiability of the cost function

As we have seen in Figure 4.7, the cost function has points where it is non-differentiable in the sense of that the directional gradients do not match. These points belong to a set of hyperplanes, which split the domain D_b and pass through the minima of the cost function. Therefore, it will force us to discard other classical minimization techniques.

The domain of the extraction vector was given in (4.52). However, it can be partitioned into two complementary sets according to the differentiability of the cost function,

$$D_{\mathbf{u}} = D_{\mathbf{u}}^d + D_{\mathbf{u}}^{nd}. \quad (4.75)$$

- There is a set of extraction vectors for which the cost function is differentiable:

$$D_{\mathbf{u}}^d = \left\{ \mathbf{u} \in D_{\mathbf{u}} : \prod_{i=1}^n |g_i| > 0 \right\}, \quad (4.76)$$

where the magnitudes and the phases are defined in (4.5) and (4.6), respectively. Indeed, this set contains all the vectors in the interior of the

convex cone $C_{\mathbf{u}}(\boldsymbol{\theta})$.

- There is also a complementary set where the cost function is non-differentiable, and whose vectors belong to the boundary $\partial C_{\mathbf{u}}(\boldsymbol{\theta}), \forall \boldsymbol{\theta}$:

$$D_{\mathbf{u}}^{nd} = \left\{ \mathbf{u} \in D_{\mathbf{u}} : \prod_{i=1}^n |g_i| = 0 \right\}. \quad (4.77)$$

Although the set $D_{\mathbf{u}}^{nd}$ contains the minima of the cost function, in practice, this set has zero volume. Therefore, we can assume that the cost function is fully differentiable at all the extraction vectors that we can find in practice.

4.7.3 A gradient descent algorithm

To minimize the cost function $\psi(\mathbf{u})$, we propose the typical update equation of a gradient descent algorithm,

$$\mathbf{u} \leftarrow \mathbf{u} - \mu \nabla_{\mathbf{u}} \psi(\mathbf{u}). \quad (4.78)$$

Note how the cost function depends on \mathbf{u} and on the whitened observations \mathbf{z} , so they play the same role as \mathbf{b} and \mathbf{x} , respectively.

Using the rules of complex derivation,

$$\nabla_{\mathbf{u}^*} \psi(\mathbf{u}) = \nabla_{\mathbf{u}^*} \left(\frac{L(\mathbf{u}^H \mathbf{z})}{\|\mathbf{u}\|} \right) = \frac{1}{\|\mathbf{u}\|^2} \left(\|\mathbf{u}\| \nabla_{\mathbf{u}^*} L(\mathbf{u}^H \mathbf{z}) - L(\mathbf{u}^H \mathbf{z}) \nabla_{\mathbf{u}^*} \|\mathbf{u}\| \right). \quad (4.79)$$

It is known that the gradient of a 2-norm is

$$\nabla_{\mathbf{u}^*} \|\mathbf{u}\| = \nabla_{\mathbf{u}^*} (\mathbf{u}^H \mathbf{u})^{1/2} = \frac{\mathbf{u}}{2(\mathbf{u}^H \mathbf{u})^{1/2}} = \frac{\mathbf{u}}{2\|\mathbf{u}\|}, \quad (4.80)$$

so the gradient of the contrast depends on the gradient of the convex perimeter,

$$\begin{aligned} \nabla_{\mathbf{u}^*} \psi(\mathbf{u}) &= \frac{1}{\|\mathbf{u}\|^2} \left(\|\mathbf{u}\| \nabla_{\mathbf{u}^*} L(\mathbf{u}^H \mathbf{z}) - L(\mathbf{u}^H \mathbf{z}) \frac{\mathbf{u}}{2\|\mathbf{u}\|} \right) \\ &= \frac{1}{\|\mathbf{u}\|} \left(\nabla_{\mathbf{u}^*} L(\mathbf{u}^H \mathbf{z}) - \psi(\mathbf{u}) \frac{\mathbf{u}}{2\|\mathbf{u}\|} \right). \end{aligned} \quad (4.81)$$

4.7.4 Practical computation of the convex perimeter

One important issue when moving from the theory to the implementation stage is the practical estimation of some values. As some of the quantities are complex-valued random vectors, we only have available some samples drawn from stochastic processes. Therefore, it is important to estimate them as accurate as we can.

In our particular case, we need to estimate the convex perimeter $L(y)$, from the support of the sample density S_y of the realizations. For this reason, BCA fits better in those applications where the extreme points of the convex support have a significant chance to occur. That is because the sample convex perimeter depends only on those extreme points laying on the bound of the set.

Suppose that the length of the sequences of samples is denoted by T . Let us denote the available sequence of whitened observations by $\{\mathbf{z}(1), \dots, \mathbf{z}(T)\}$, so the output sequence is

$$\{y(k)\} = \mathbf{u}^H \{\mathbf{z}(k)\}, \quad k = 1, \dots, T. \quad (4.82)$$

The first step is to compute the convex hull of the output, $\text{conv}\{y\}$, from the samples of the output sequence. In computation geometry, numerous algorithms have been proposed for computing the convex hull of the elements of a sequence $\{y(k)\}$ of T points (see, for instance, (Preparata, 1977)).

These algorithms usually return an array of indices $[\varsigma_0 \ \varsigma_1 \ \dots \ \varsigma_V]$. This array is circular, in the sense that the first and the last indices are the same, $\varsigma_0 = \varsigma_V$. This array determines the position of the vertices of the convex hull of the output, in clockwise order:

$$\text{conv}\{y\} = \{y(\varsigma_0), \dots, y(\varsigma_V)\}. \quad (4.83)$$

The *edges* between consecutive vertices of the convex hull of the output are

$$\Delta y_i = y(\varsigma_i) - y(\varsigma_{i-1}) \in \mathbb{C}, \quad (4.84)$$

where $i = 1, \dots, V$ is the index that moves along all the vertices of the convex hull of the output. It can be seen that the convex perimeter of this set is the summation of the lengths of all these edges,

$$L(y) = \sum_{i=1}^V |\Delta y_i|. \quad (4.85)$$

Example 4.1 illustrates the previous concepts for a simple set of samples.

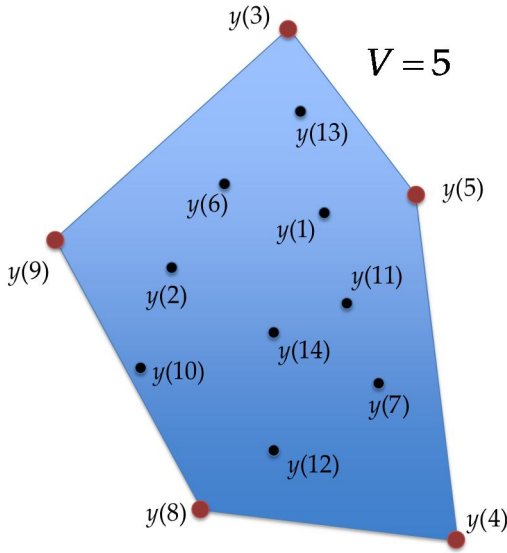


Figure 4.11 - A set of samples from the output sequence y , with $T=14$. Observe how there are only $V=5$ vertices in the convex hull.

Example 4.1 Let us observe Figure 4.11, where we have a set of samples in the complex plane, drawn from an output sequence $\{y(k)\}$, $k=1, \dots, 14$. If we compute its convex hull, it returns the indices of the extreme points of the set,

$$\text{conv}\{y\} = [\zeta_0, \dots, \zeta_V] = [3, 5, 4, 8, 9, 3]. \quad (4.86)$$

Thus, the edges between consecutive vertices of the output are

$$\begin{aligned} \Delta y_1 &= y(\zeta_1) - y(\zeta_0) = y(5) - y(3), \\ \Delta y_2 &= y(\zeta_2) - y(\zeta_1) = y(4) - y(5), \\ \Delta y_3 &= y(\zeta_3) - y(\zeta_2) = y(8) - y(4), \\ \Delta y_4 &= y(\zeta_4) - y(\zeta_3) = y(9) - y(8), \\ \Delta y_5 &= y(\zeta_5) - y(\zeta_4) = y(3) - y(9). \end{aligned} \quad (4.87)$$

And finally, the convex perimeter is given by

$$L(y) = |\Delta y_1| + |\Delta y_2| + |\Delta y_3| + |\Delta y_4| + |\Delta y_5| \in \mathbb{R}. \quad (4.88)$$

4.7.5 The gradient of the normalized convex perimeter

The *normalized convex perimeter* of the output is obtained by sequentially accumulating the modulus of consecutive differences, and dividing by the 2-norm of the extraction vector,

$$\psi(\mathbf{u}) = \frac{1}{\|\mathbf{u}\|} \sum_{i=1}^V |\Delta y_i|. \quad (4.89)$$

We use the gradient of the contrast (4.89) in the minimization algorithm.

Theorem 4.8 (Gradient of the normalized convex perimeter)

Given the cost function of the normalized convex perimeter (4.89), the gradient needed for the update of the gradient descent algorithm (4.78) is computed as

$$\nabla_{\mathbf{u}^*} \psi(\mathbf{u}) = \frac{1}{2\|\mathbf{u}\|} \sum_{i=1}^V \left(\left(\Delta \mathbf{z}_i - \frac{\mathbf{u} \Delta y_i}{\|\mathbf{u}\|^2} \right) \frac{\Delta y_i^*}{|\Delta y_i|} \right). \quad (4.90)$$

The segments Δy_i and $\Delta \mathbf{z}_i$ are computed from (4.84) and $\Delta \mathbf{z}_i = \mathbf{z}(\zeta_i) - \mathbf{z}(\zeta_{i-1})$, respectively. Note how the only convex hull needed is $\text{conv}\{y\}$.

Proof: Starting at (4.81) and by substituting (4.89),

$$\begin{aligned} \nabla_{\mathbf{u}^*} \psi(\mathbf{u}) &= \nabla_{\mathbf{u}^*} \left(\frac{1}{\|\mathbf{u}\|} \sum_{i=1}^V |\Delta y_i| \right) \\ &= \frac{1}{\|\mathbf{u}\|} \left(\sum_{i=1}^V \nabla_{\mathbf{u}^*} (|\Delta y_i|) - \frac{\mathbf{u}}{2\|\mathbf{u}\|^2} \sum_{i=1}^V |\Delta y_i| \right). \end{aligned} \quad (4.91)$$

The gradient of the convex perimeter in the first term of (4.91) can be derived by applying Wirtinger calculus on the moduli $|\Delta y_i|$,

$$\begin{aligned} \nabla_{\mathbf{u}^*} (|\Delta y_i|) &= \nabla_{\mathbf{u}^*} (\Delta y_i \Delta y_i^*)^{1/2} \\ &= \nabla_{\Delta y_i^*} (|\Delta y_i|) \underbrace{\nabla_{\mathbf{u}^*} (\Delta y_i^*)}_0 + \nabla_{\Delta y_i} (|\Delta y_i|) \nabla_{\mathbf{u}^*} (\Delta y_i). \end{aligned} \quad (4.92)$$

The first term is zero, because the segments (4.84) applied to the model (4.82) gives

$$\Delta y_i = \mathbf{u}^H \Delta \mathbf{z}_i, \quad (4.93)$$

$$\Delta y_i^* = \mathbf{u}^T \Delta \mathbf{z}_i^*. \quad (4.94)$$

Therefore, using (4.93) the gradient of the modulus results

$$\nabla_{\mathbf{u}^*} (|\Delta y_i|) = \frac{\Delta y_i^*}{2(\Delta y_i \Delta y_i^*)^{1/2}} \nabla_{\mathbf{u}^*} (\Delta y_i) = \frac{\Delta y_i^*}{2|\Delta y_i|} \Delta \mathbf{z}_i. \quad (4.95)$$

And finally, substituting the gradient (4.95) in (4.91),

$$\begin{aligned} \nabla_{\mathbf{u}^*} \psi(\mathbf{u}) &= \frac{1}{\|\mathbf{u}\|} \sum_{i=1}^V \left(\frac{\Delta y_i^*}{2|\Delta y_i|} \Delta \mathbf{z}_i - \frac{\mathbf{u} |\Delta y_i|}{2\|\mathbf{u}\|^2} \right) \\ &= \frac{1}{2\|\mathbf{u}\|} \sum_{i=1}^V \left(\Delta \mathbf{z}_i - \frac{\mathbf{u} \Delta y_i}{\|\mathbf{u}\|^2} \right) \frac{\Delta y_i^*}{|\Delta y_i|}. \end{aligned} \quad (4.96)$$

■

One can see that the gradient only depends on the extraction vector \mathbf{u} , and on the segments Δy_i and $\Delta \mathbf{z}_i$. At each iteration, all these quantities are easily computed on an iterative algorithm.

4.8 Blind Source Extraction and Separation by using BCA

This section presents the most direct application of BCA: the recovery of one or more sources from an unknown linear mixture. The models were given in Section 4.2.1, at the beginning of this Chapter.

4.8.1 A BCA algorithm for the Blind Source Extraction

In Table 4.1, the pseudo-code of a function which implements the extraction algorithm is presented. Note how the extraction vector is normalized at the end of each iteration. That is because any implementation will run with finite precision, so the division by the 2-norm helps the stability of the iterations.

It is important to remark that the segments $\Delta \mathbf{z}_i$ do not need the computation of the convex hull of any of the whitened observations contained in the vector \mathbf{z} . Looking into its definition, it can be seen that they only depend on certain samples of the vector \mathbf{z} . These selected samples are the ones that yield to the samples on the convex hull of the output, $\{y(\zeta_0), \dots, y(\zeta_V)\}$.

<i>Inputs</i>	\mathbf{z} : whitened observations		
<i>Repeat until convergence</i>	$\mathbf{y} = \mathbf{u}^H \mathbf{z}$: estimate of the source		
	<i>Computation of the gradient</i>	$[\zeta_0, \dots, \zeta_V] = \text{conv}\{y\}$	
		<i>Repeat for</i> $i = 1, \dots, V$	$\Delta \mathbf{z}_i = \mathbf{z}(\zeta_i) - \mathbf{z}(\zeta_{i-1})$ $\Delta y_i = y(\zeta_i) - y(\zeta_{i-1})$
		$\nabla_{\mathbf{u}} \psi(\mathbf{u}) = \frac{1}{2\ \mathbf{u}\ } \sum_{i=1}^V \left(\left(\Delta \mathbf{z}_i - \frac{\mathbf{u} \Delta y_i}{\ \mathbf{u}\ ^2} \right) \frac{\Delta y_i^*}{ \Delta y_i } \right)$	
μ : adaptive step size $\mathbf{u} \leftarrow \mathbf{u} - \mu \nabla_{\mathbf{u}} \psi(\mathbf{u})$: update $\mathbf{u} \leftarrow \mathbf{u} / \ \mathbf{u}\ $: normalization $k \leftarrow k + 1$: increase the iteration index			
<i>Outputs</i>	\mathbf{u} : extraction vector $\mathbf{y} = \mathbf{u}^H \mathbf{z}$: final estimate of the source		

Table 4.1: Pseudo-code of a BCA extraction algorithm.

For completeness, a MatLab implementation of the BCA extraction algorithm is also available in (Cruces, 2008).

Computer simulations

We consider an instantaneous mixture with $n=3$ sources and $m=3$ observations. The mixing coefficients are drawn independently from a Gaussian random process, thus characterizing a flat Rayleigh fading channel. Suppose that the model does not change during at least $T=200$ samples. The source signals are QPSK constellations with unit power, and the SNR is set to 30 dB.

The Figure 4.12 shows the state of the extraction algorithm at two different iterations. It can be seen that the global transfer vector tends towards $[0 \ 0 \ 1]^T$, which corresponds to a successful extraction of the 3rd source. This can also be corroborated by identifying the shape of the constellation at the output.

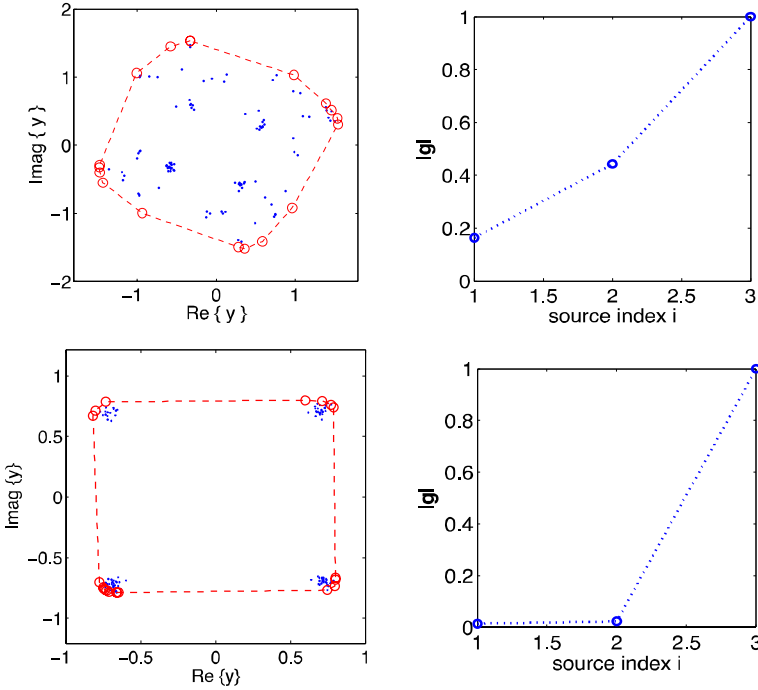


Figure 4.12 - This figure shows, on the left column, the output of the extraction algorithm $y(k)$. The red circles are the samples in the complex plane that belongs to the convex hull, and the edges $\Delta y_{(i)}$ are represented by a dashed line. On the right column, the absolute value of the global transfer vector $|g|$ is shown. The upper row shows the state of the extraction at the 4-th iteration, and the lower row shows the state at the 40-th iteration.

4.8.2 Computational complexity

Assume that the number of samples available T is much greater than the number of sensors m , which is greater or equal than the number of sources n .

$$T \gg m \geq n. \quad (4.97)$$

Under this assumption, the computational complexity of the whitening (4.71) depends on

- $O(m^2T)$, for the computation of the correlation matrix.
- $O(m^3)$, for the eigenvalue decomposition.
- $O(nmT)$, for the final matrix products.

So (4.97) implies that the overall computational cost of whitening is $O(m^2T)$.

For the computation of a planar convex hull of a finite set of T samples, the complexity rises to $O(T \log(T))$ in the worst case¹³ (Graham, 1972). However, if one takes into account the number of vertices V of $\text{conv}\{y\}$, there are better practical algorithms.

These algorithms, whose complexity depends on both T and V , are called *output-sensitive*. For the worst case in this family of algorithms (Chan, 1996), the computational complexity is $O(T \log(V))$.

The number of vertices V is usually much smaller than T . Two approximations can be done here, depending on the value of the SNR (*Signal to Noise Ratio*):

- At high SNRs, V can be approached by the sum of cardinalities of the sets of vertices of the convex cover of each one of the sources.

$$V \approx V_{high} = \sum_{i=1}^n \left| \text{ext}(\text{conv}\{s_i\}) \right|. \quad (4.98)$$

- At low SNRs, V is biased towards

$$V \approx V_{low} = 8\sqrt{\pi}, \quad (4.99)$$

which is a tight upper-bound on the expected number of vertices of the convex hull of the output when it is dominated by the contribution of a complex Gaussian noise.

Therefore, the expected complexity per iteration of the BCA extraction algorithm can be roughly approached by

- At high SNRs, $O(T \log(V_{high}))$.
- At low SNRs, $O(T \log(\max\{V_{high}, V_{low}\}))$.

Nevertheless, the computational complexity of the BCA extraction algorithm is limited by the calculation of the convex hull of the output sequence.

4.8.3 Blind Source Separation

The next step is to implement a BSS method, based on the previous BSE algorithm. It is relatively easy to extend the algorithm to the separation of all the bounded sources. This application is presented in (Aguilera, 2012). After the

¹³ This is the same order of complexity of the sorting of a list with T elements.

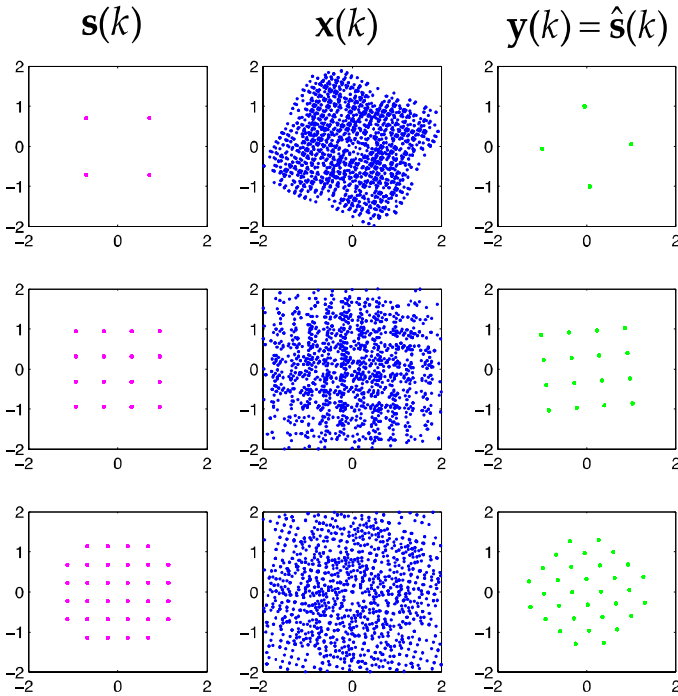


Figure 4.13 – First experiment. Example of the blind separation of sources by using a BCA extraction and deflation algorithm. The first column shows the original sources: a QPSK, a 16-QAM and a 32-QAM constellations. The second column shows the observations obtained after the mixing. Finally, the third column shows the extraction of the three original sources after a sequential extraction and deflation procedure.

extraction of the first component of the mixture, one has to deflate the contribution of the estimated source from the observations, so as to obtain a new mixture of $n-1$ sources.

Then, as described in (Cruces, 2004) or (Delfosse, 1995), the extraction and deflation processes alternate until the desired number of sources are recovered. Although the source processes are spatially uncorrelated, their finite sample versions are usually slightly correlated due to the limited number of observations.

Thus, it is usually recommended a second round of updates of the source extraction vectors which relaxes their exact orthogonality, improving in this way, the final performance of the separation algorithm.

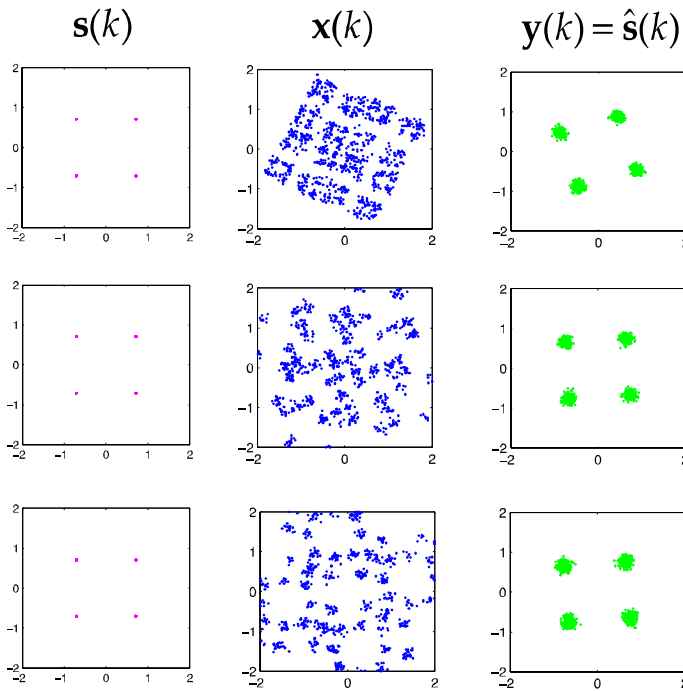


Figure 4.14 – Second experiment. Example of the blind separation of dependent sources by using a BCA extraction and deflation algorithm. The first column shows the three QPSK original sources. The second column shows the noisy observations obtained after the mixing. The third column shows the extraction of the original sources after a sequential extraction and deflation procedure.

Computer simulations

We run three experiments to illustrate how the BSS methodology provides accurate estimations of the original sources.

- In the first experiment, we use three sources whose shapes are constellations used in the *Quadrature Amplitude Modulation (QAM)* communications scheme. We use the same parameters of the previous BSE experiment, but this time we use 200 iterations and a sample length of $T = 1000$ symbols. This is due to the more complex structure of the desired signals. Also, there is no additive noise in this experiment.

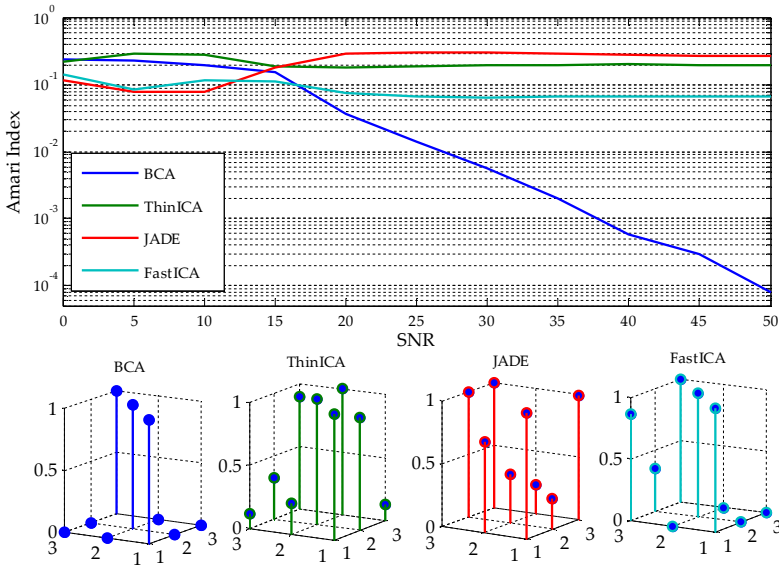


Figure 4.15 – Third experiment. This figure shows the result of executing the BSS methods based on BCA, ThinICA, JADE, and FastICA to a noisy mixture of dependent sources. (upper row) Comparison of the Amari Index (AI) versus the SNR. (lower row) Each one of the points corresponds to the absolute value of the coefficients of the global transfer vector $|\mathbf{G}|$, for each one of the compared methods, for a SNR=50 dB situation. The permutation and the scale is corrected to show the successful extraction when the matrix is close to the identity.

In Figure 4.13, the result of this experiment is shown. Note how the three sources are successfully extracted, up to permutations and ambiguities in the phase of the constellation. The lack of noise in this experiment helps the convergence and the successful separation of the whole set of sources.

- In the second experiment, three QPSK dependent sources (with correlation coefficient of 0.6) are mixed with the same mixture matrix that in the previous examples. However, this time a AWGN noise is added to the observations, so the recovery may not be perfect. The SNR is 20 dB, the sample length is $T = 500$ symbols, and the number of iterations can be reduced to 50.

The Figure 4.14 shows the result of this separation. The three sources are extracted even if they are dependent, but this time their constellations are contaminated by the noise. Again, we have the same ambiguities on the order and on the phase of the signals recovered.

- In the third experiment, we mix three QPSK sources and obtain three noisy observations (SNR goes from 0 to 50 dB). This time, we compare the performance of the proposed method with other BSS techniques like ThinICA (Cruces, 2004), JADE (Cardoso, 1997) or FastICA (see (Hyvärinen, 1997) and its complex version (Bingham, 2000)). The number of samples is set to $T = 1500$ samples, and dependent sources (with a correlation coefficient of 0.5) are considered.

The result of this comparison is shown in Figure 4.15, for the absolute value of the global transfer matrix \mathbf{G} . The *Amari Index (AI)* (Amari, 1996) is used to measure the quality of the separation, defined for a square matrix as,

$$AI(\mathbf{G}) = \frac{1}{n^2} \sum_{j=1}^n \left(\frac{\sum_{i=1}^n |G_{ji}|^2}{\max\{\|\mathbf{g}_{j\cdot}\|^2\}} - 1 \right), \quad (4.100)$$

where $\mathbf{g}_{j\cdot}$, $j = 1, \dots, n$, is the j -th row of the matrix \mathbf{G} . The Amari Index is zero in a situation of perfect separation.

It can be seen that, when the sources are dependent, the main hypothesis for ICA is violated and the source separation is not possible for ThinICA, JADE, and FastICA. Nevertheless, the proposed BCA algorithm is the only one able to recover the original signals in medium-high SNR scenarios, as long as they satisfy the conditions seen in Section 4.3.

4.9 Latent Component Decomposition by using BCA

During this entire chapter, we have explained the theory of Bounded Component Analysis by using the application of blind source extraction. However, as we will see in Chapter 6, there is also the possibility to apply the tools presented to solve another type of problems. In this section, we briefly review an application related to the topic of BSS and BSE, but with a different model.

4.9.1 Linear model

Suppose that we want to decompose a given observations vector (see (Cruces, 2010)) into a set of n bounded *latent components* $\mathbf{x}_i(k)$,

$$\mathbf{x}(k) = \sum_{i=1}^n \mathbf{x}_i(k) \in \mathbb{C}^m. \quad (4.101)$$

Note that this model can be directly converted into a noiseless linear mixture model like (4.1) by setting the structure:

$$\mathbf{x}_i(k) = \mathbf{a}_i s_i(k), \quad i = 1, \dots, n. \quad (4.102)$$

The linear independent vectors $\mathbf{a}_i \in \mathbb{C}^m$ determine the linear subspaces of the components, and the random process $s_i(k)$ are hidden factors (sources) that account for the statistical structure and the geometry of the components.

The main assumptions for the decomposition are the same that in Section 4.3:

- The non-degeneracy of the component vectors, $\mathbf{x}_i(k)$ with bounded supports $S_{\mathbf{x}_i(k)} \subset \mathbb{C}^m$.
- The decomposition of the convex support of the observations $S_{\mathbf{x}(k)}$ as a Minkowski direct sum of the convex supports of the component signals,

$$S_{\mathbf{x}(k)} = S_{\mathbf{x}_1(k)} \oplus \dots \oplus S_{\mathbf{x}_i(k)}. \quad (4.103)$$

Note that this second assumption automatically enforces a limitation in the number of component signals n , which should be always equal or smaller than the number of observations, ($n \leq m$).

The identifiability, uniqueness and indeterminacies detailed in Section 4.4 also hold for this model, by using the relation (4.102). Thus, the bounded component decomposition is unique up to permutations in the order of the components.

Without loss of generality, hereinafter, we assume that the component vectors $\mathbf{x}_i(k)$ are further indecomposable, in the sense that each convex support $S_{\mathbf{x}_i(k)}$, cannot be further decomposed as a Minkowski direct sum of any other complex sets with nonzero dimension. This is not a restriction, since one can always replace any decomposable set $S_{\mathbf{x}_i(k)}$ in the right-hand side of (4.103), as a direct sum of indecomposable components.

<i>Inputs</i>	\mathbf{x} : observations		
<i>For each source i</i>	$\mathbf{z} = \mathbf{W}\mathbf{x}$: whiten the observations $y_i = \mathbf{u}_i^H \mathbf{z}$: estimate of the source by using BCA (Table 4.1)		
	<i>Identify the corresponding mixing vector</i>	<i>Repeat for $j=1, \dots, m$</i>	$[\mathbf{a}_i]_j = \arg \left\{ \min_{\alpha_j} \left\{ L(x_j - \alpha_j y_i) \right\} \right\}$
	<i>Identify the i-th latent component</i>	$\mathbf{x}_i = \mathbf{a}_i y_i$	
	<i>Eliminate the contribution</i>	$\mathbf{x} \leftarrow \mathbf{x} - \mathbf{x}_i$	
	$n = n - 1$: dimensional reduction $i = i + 1$: go to the next source		
<i>Outputs</i>	$\mathbf{a}_1, \dots, \mathbf{a}_n$: mixing vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$: the set of bounded components		

Table 4.2: Pseudo-code of a bounded component decomposition algorithm.

4.9.2 The sequential recovery of the components

In Table 4.2, we present an algorithm for the sequential recovery of the components of the observations. The main steps of this recovery consist on exploiting the BCA assumptions for developing a blind method for the identification of a bounded source and of its corresponding mixing vector.

The i -th source can be identified by applying the extraction vector \mathbf{u}_i to the observations,

$$y_i(k) = \mathbf{u}_i^H \mathbf{z}(k) = \hat{s}_i(k), \quad i = 1, \dots, n. \quad (4.104)$$

This output is an estimate of the source corresponding to the local minimum at the bottom of the basin where the algorithm is initialized. The next step is to estimate the column \mathbf{a}_i , needed to obtain the latent component (4.102).

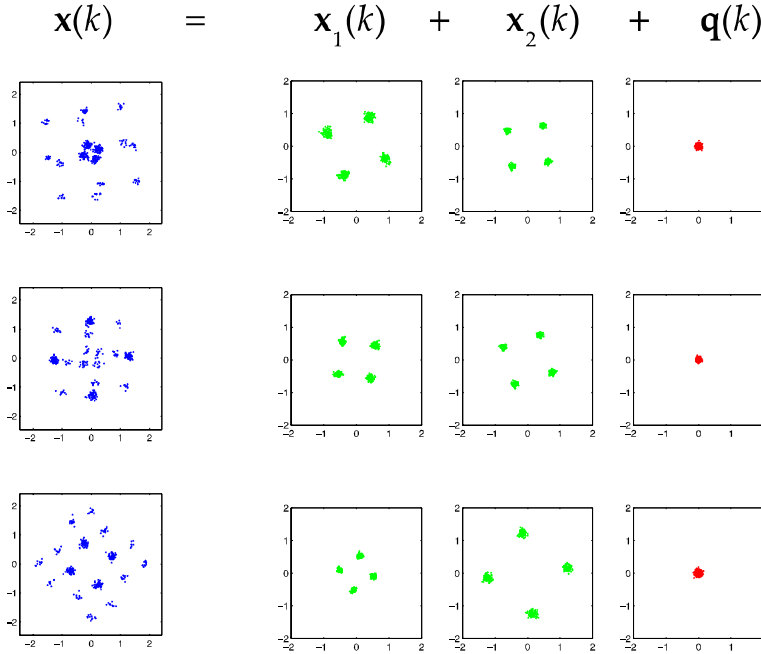


Figure 4.16 – First experiment of the Latent Component Decomposition. This figure shows the result of a bounded component decomposition applied to a set of three observations, marked by blue dots. The bounded components are marked by green dots, while the residual unbounded component is marked by red dots.

For the identification of the mixing vector, we propose the following estimate¹⁴:

$$[\mathbf{a}_i]_j = \arg \left\{ \min_{\alpha_j} \left\{ L(r_j(k)) \right\} \right\}, \quad j = 1, \dots, m, \quad (4.105)$$

where $r_j(k) = x_j(k) - \alpha_j y_i(k)$ is the *remainder* which is obtained when the current estimated component is subtracted from the observations.

These elements are chosen in such a way that they minimize the convex perimeter of the coordinates of the residual of the approximation. Of course, the optimization problems (4.105) can also be minimized via a gradient descent algorithm.

¹⁴ This vector can also be estimated by using a MMSE approach. That is, $\mathbf{a}_i = \mathbf{r}_{xy_i} / \sigma_{y_i}^2$, where \mathbf{r}_{xy_i} is the temporal cross-correlation between the observations and the recovered source.

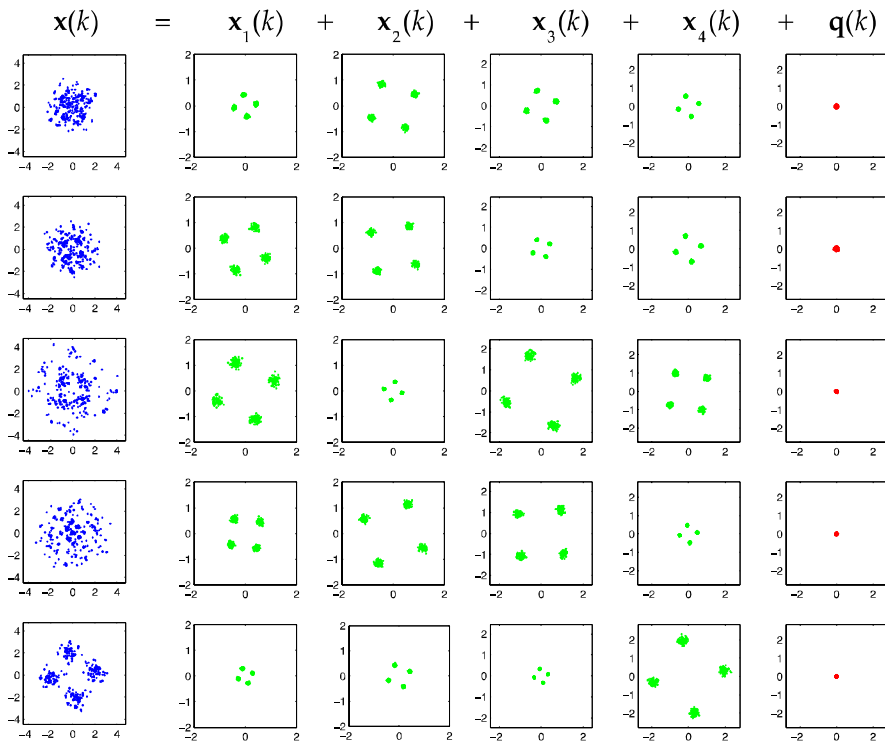


Figure 4.17 – Second experiment of the Latent Component Decomposition. This figure shows the result of a bounded component decomposition applied to a set of five observations, marked by blue dots. The bounded components are marked by green dots, while the residual unbounded component is marked by red dots.

4.9.3 Computer simulations

As in the previous section, computer simulations were done to show the performance of the method presented. We derive three experiments to help the reader to understand the sequential recovery of the components.

- In the first experiment, we aim to decompose $m = 3$ observations into $n = 2$ bounded components plus a noise component $\mathbf{q}(k)$, with $\text{SNR} = 20$ dB. This residual term accounts for the part of the observations that cannot be explained by bounded parts. It can be done with only

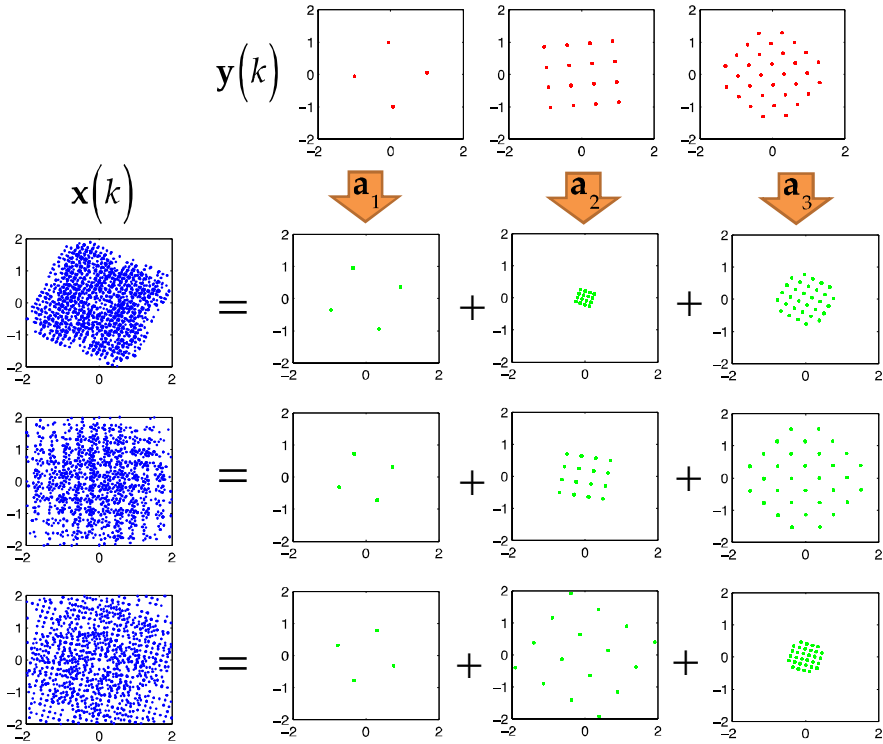


Figure 4.18 – Third experiment of the Latent Component Decomposition. The blue dots represent the three observations, while the green dots represent the bounded components decomposed sequentially by the algorithm presented in Table 4.2. At the top of the figure, the sources corresponding to the output of (4.104) are marked in red. The extraction vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are responsible for the conformation of the bounded components.

100 iterations for each source extraction and for the estimation of the coefficients given by (4.105). The sample length was set to $T = 200$ symbols.

In the Figure 4.16 we illustrate the decomposition resulted from this experiment. The two components recovered have bounded support, while the unbounded component may have samples arbitrarily large.

- The second experiment is similar to the first one, but with higher dimensions. This time, we set $m = 5$ and $n = 4$. Due to the higher dimensionality of the problem, we need 200 iterations and a length of $T = 400$ symbols.

Figure 4.17 shows the four bounded components that decomposes the observations. Note how even in high dimensionality scenarios, the sequential extraction and deflation algorithm is able to converge.

- The third and last experiment on this chapter aims to explain the tools used in the deflation algorithm to extract sequentially each of the components. This time, we work in a noiseless scenario, where the three observations come again from the field of communications: a QPSK, a 16-QAM, and a 32-QAM constellation. Due to the complex geometry of these sources, a sample length of $T = 2000$ symbols is used. 200 iterations are needed for the extraction of each source and for the identification of the components defined by (4.105).

In the Figure 4.18, an illustration of the overall method is shown. The first source is extracted by using the original BCA algorithm. After that, the mixing vector \mathbf{a}_1 is estimated by minimizing the contribution of the resulting bounded component to the observations, as explained in (4.105). The bounded component resulted from this process is then subtracted from the observations.

This process is repeated for all the sources, decreasing the dimension of the source set after each extraction and decomposition. Note how the method is equivalent to the sequential BSS extraction presented in Section 4.8.3, but in this case we are not interested in the sources and only use them as an intermediate step.

4.10 Conclusions

In this chapter, we have presented the theory of Bounded Component Analysis, a framework for source separation and extraction where the main assumption is the bounded nature of the support of the sources.

It is based upon the Minkowski-Brunn theory of convex bodies, and mixes tools from convex geometry and time series analysis into a unified framework. Some of the most important results for this chapter are:

- For bounded sources, the main assumptions in BCA are weak and easily satisfied in practical situation. They are the compactness and non-degeneracy of the sources, the Cartesian decomposition of the convex support of the sources and the full column-rank of the mixing matrix.
- The support of the observations can be decomposed as a Minkowski direct sum of convex summands, each one representing the support of

one source multiplied by one column of the mixing matrix.

- The decomposition of the support of the observations is unique, up to permutation of the direct summands. There are also scaling and phase indeterminacies on the sources, as in other blind source separation algorithms.
- The criterion of minimum normalized convex perimeter establishes that the minimization of the perimeter of the output, divided by the norm of the extraction vector, always leads to valid solutions of the source extraction problem.
- However, the number of samples required to accurately estimate the convex perimeter may be prohibitive if the sources do not have clear boundaries.

We presented also some of the main steps for the practical optimization of the cost function. After presenting how to whiten the observations, the gradient of the contrast was derived, based upon the rules of Wirtinger calculus. A pseudo-code with a practical implementation of the BCA extraction algorithm was included, with all the intermediate computations needed to build up the gradient. Finally, a discussion on the computational complexity of such algorithm concludes that the most demanding task is the calculation of the convex hull of the output sequence.

In addition, a blind source separation method and latent component decomposition method are outlined. They are based on the sequential recovery of the sources and the components, by estimating the mixing vectors and minimizing their contribution to the observation set. It was shown that the recovery of the sources is possible even if they are somehow dependent.

It is important to understand that this chapter has been built assuming a scenario where the objective is to blindly recover one or more sources from an unknown mixture. That was because one needs to focus on a concrete application to deep into the properties, behaviors and solutions of the theory. However, the BCA framework can be easily extended to supervised problems, as we will see in Chapter 6.

5 CONVERGENCE STUDY OF A BCA EXTRACTION ALGORITHM

It is not the mountain we conquer, but ourselves.

- Sir Edmund Hillary -

The global convergence of an optimization algorithm deals with the ability of reaching one of the minima of the cost function, with independence of the initialization point where it starts. It is a very desirable feature in problems where there are not erroneous solutions. We aim to make this analysis as general as possible, covering the possibility of having sources in the mixture from different constellations and different perimeters.

In a noiseless situation, our convergence analysis provides recommendations for setting the step size of the algorithm so as guarantee the *global monotonous convergence* to the source that is closest to the initialization of the algorithm (in a given sense) while keeping, at the same time, a fast local convergence rate in the neighborhood of this solution.

5.1 Introduction

In this chapter, we address the study of the global monotonous convergence of the extraction algorithm proposed in Chapter 4. This property does not rely only in the absence of erroneous minima in the criterion (see Theorem 4.7), but also on the global stability of the learning rules.

The global monotonous convergence is a very desirable feature, however, in general it does not hold for most of the existing component analysis algorithms, up to some notable exception like (Shalvi, 1994 pp. 121-180), (Papadias, 2000) and (Erdogan, 2007), among others.

The chapter is organized as follows. We start by reviewing the cost function and describing its shape in Section 5.2. In Section 5.3, we derive fast step sizes based on the first and second order local analysis of the cost function, although they do not guarantee the global convergence of the BCA extraction algorithm.

The study of the global monotonous convergence is derived in Section 5.4, after a deep analysis of the geometry of the update. The result is the proposal of some convenient bounds for the step size and their practical estimators, whose behavior is illustrated by means of computer simulations in Section 5.5. Finally, Section 5.6 summarizes the conclusions to this chapter.

5.2 The Special Shape of the Cost Function

In this chapter, k denotes the *iteration index*. We will use loosely the word *gradient*, which sometimes may denote the conjugate cogradient. Its meaning will be always clear from the context.

The design of appropriate step sizes starts with the analysis of the shape of the cost function. The proposed cost function for the BCA extraction algorithm is the normalized convex perimeter of the output,

$$\psi(\mathbf{u}) = \frac{L(y)}{\|\mathbf{u}\|} = \frac{L(\mathbf{u}^H \mathbf{z})}{\|\mathbf{u}\|}, \quad (5.1)$$

where $\mathbf{u} \in \mathbb{C}^n$ is the extraction vector and $\mathbf{z} \in \mathbb{C}^n$ is the vector of whitened observations.

This cost function cannot be easily visualized in the 3D space because of the higher dimensionality of its domain. In Figure 5.1, the shape of the cost function for a real-valued scenario with $n = 2$ is shown. The radial lines of the contour plot illustrate the scaling invariance of the cost function. For illustrative

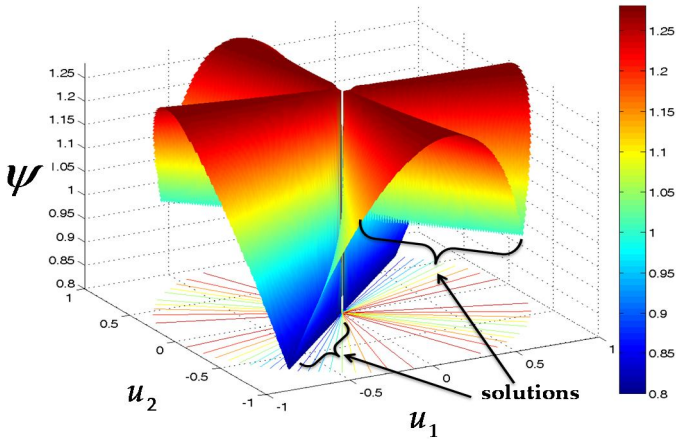


Figure 5.1 – Surface and contour plot of the cost function when considering a mixture $A = [1, 1; -1, 1]$ of two sources with respective perimeters: 1 and 0'8.

purposes, the boundary of the graph corresponds to an extraction vector of constant norm.

The graph also reveals the subspaces where the function is non-differentiable. Each one of these subspaces corresponds to the extraction of one source, with indeterminacy on the value of its magnitude and phase. In addition, it can be seen that the point of initialization is critical for the determination of the local minima attained.

One can describe its essential features imagining its opposite $-\psi(\mathbf{u})$ as a mountain landscape with edges converging toward the peaks of the mountains, which are the solutions to the problem.

From this representative view of the function, one can understand why we do not recommend setting the step size $\mu^{(k)}$ with any classical quasi-optima line-search procedure. These kinds of searches would drive the algorithm in a few steps toward the vicinity of the set of non-differentiable points $D_{\mathbf{u}}^{nd}$ (see Section 4.7.2), which represents the edge of the mountains. Once there, the subsequent quasi-optimal gradient descent updates would be progressively smaller, eventually stopping the convergence.

For this reason, ad-hoc standard optimization methods for differentiable functions usually fail to minimize it. Therefore, specialized methods are required for this task, based on adaptive step sizes.

5.3 Step Sizes Based On the Shape of the Cost Function

Part of this Section was presented in (Aguilera, 2012). To set the step size, we analyze the local influence of $\mu^{(k)}$ in the value of the cost function. The update equation of the descent algorithm was presented in Section 4.7.3,

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mu^{(k)} \nabla_{\mathbf{u}} \psi(\mathbf{u}), \quad (5.2)$$

where

$$\nabla_{\mathbf{u}} \psi(\mathbf{u}) = \frac{1}{\|\mathbf{u}\|} \left(\nabla_{\mathbf{u}} L(y) - \frac{\mathbf{u}}{2\|\mathbf{u}\|^2} L(y) \right). \quad (5.3)$$

Note that the perimeter of the output $L(y)$ and its gradient $\nabla_{\mathbf{u}} L(y)$ can both be accurately estimated from the knowledge of the extraction vector \mathbf{u} and of a sufficient large number of observations \mathbf{z} .

Provided that $\mathbf{u} \in D_{\mathbf{u}}^d$, within a neighborhood of the current point, the local behavior of the cost function is given by the Taylor approximation (see Appendix A2.1.3 on (Schreier, 2010))

$$\begin{aligned} \psi(\mathbf{u}^{(k+1)}) &= \psi(\mathbf{u}^{(k)}) + \underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u}) (\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)}) \\ &\quad + \frac{1}{2} (\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)})^H \underline{H}_{\mathbf{uu}} \psi(\mathbf{u}) (\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)}) + \dots, \end{aligned} \quad (5.4)$$

where

$$\underline{\mathbf{u}} = \begin{bmatrix} \mathbf{u} \\ \mathbf{u}^* \end{bmatrix} \quad (5.5)$$

is the augmented representation of the extraction vector,

$$\underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u}) = \begin{bmatrix} \nabla_{\mathbf{u}} \psi(\mathbf{u}) \\ \nabla_{\mathbf{u}^*} \psi(\mathbf{u}) \end{bmatrix} \in \mathbb{C}^{2n} \quad (5.6)$$

is the *augmented gradient* of the cost function, and

$$\underline{H}_{\mathbf{uu}} \psi(\mathbf{u}) = \begin{bmatrix} \mathbf{H}_{\mathbf{u}} & \tilde{\mathbf{H}}_{\mathbf{u}} \\ \tilde{\mathbf{H}}_{\mathbf{u}}^* & \mathbf{H}_{\mathbf{u}^*} \end{bmatrix} \in \mathbb{C}^{2n \times 2n}, \quad (5.7)$$

is the *augmented Hessian*, whose elements are the Hessians,

$$\mathbf{H}_{\mathbf{u}}\psi = \nabla_{\mathbf{u}} \left(\nabla_{\mathbf{u}} \psi(\mathbf{u}) \right)^T \in \mathbb{C}^{n \times n}, \quad (5.8)$$

$$\tilde{\mathbf{H}}_{\mathbf{u}}\psi = \nabla_{\mathbf{u}^*} \left(\nabla_{\mathbf{u}^*} \psi(\mathbf{u}) \right)^T \in \mathbb{C}^{n \times n}. \quad (5.9)$$

From the expression (5.4), we can derive different step sizes based upon the first or second-order approximation of the Taylor expansion.

5.3.1 The Newton-Raphson step size

The *Newton-Raphson (NR)* step size is presented in the following Theorem.

Theorem 5.1 (The Newton-Raphson Step Size)

The Newton-Raphson step size for the gradient descent algorithm is given by

$$\mu_{NR}^{(k)} = \frac{\Delta^{(k)}}{2 \left\| \nabla_{\mathbf{u}^*} \psi(\mathbf{u}^{(k)}) \right\|^2}, \quad (5.10)$$

where

$$\Delta^{(k)} = \psi(\mathbf{u}^{(k)}) - \psi_{\min} \quad (5.11)$$

is a nonnegative parameter that indicates how close is the algorithm to the convergence.

Proof: The proof of this Theorem comes from the first-order local approximation of (5.4),

$$\psi(\mathbf{u}^{(k+1)}) = \psi(\mathbf{u}^{(k)}) + \nabla_{\mathbf{u}} \psi(\mathbf{u}) \left(\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)} \right). \quad (5.12)$$

Let us assume that $\mathbf{u}^{(k)}$ belongs to the domain of the convergence of a local solution $\mathbf{u}^{(\infty)}$, whose contrast function value is $\psi_{\min} = \psi(\mathbf{u}^{(\infty)})$. Then, the rate of convergence of the gradient algorithm can be improved by finding the step size $\mu^{(k)}$ that implements the Newton-Raphson method to find the zeros of the function $\psi(\mathbf{u}^{(k+1)}) - \psi_{\min}$.

By substituting the augmented version of the update equation, i.e.,

$$\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)} = -\mu^{(k)} \nabla_{\mathbf{u}^*} \psi(\mathbf{u}), \quad (5.13)$$

and by setting $\psi(\mathbf{u}^{(k+1)}) = \psi_{\min}$ in the linear expansion (5.12),

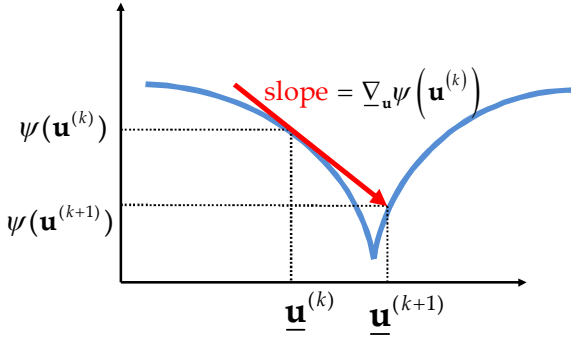


Figure 5.2 - This is a representation of the first-order local approximation used in (5.12) to derive the Newton-Raphson step size.

$$\begin{aligned}\psi_{\min} &= \psi(\mathbf{u}^{(k)}) - \mu^{(k)} \nabla_{\mathbf{u}} \psi(\mathbf{u}) \nabla_{\mathbf{u}}^* \psi(\mathbf{u}) \\ &= \psi(\mathbf{u}^{(k)}) - 2\mu^{(k)} \|\nabla_{\mathbf{u}}^* \psi(\mathbf{u})\|^2.\end{aligned}\quad (5.14)$$

Solving for $\mu^{(k)}$ yields the following NR step-size,

$$\mu_{NR}^{(k)} = \frac{\psi(\mathbf{u}^{(k)}) - \psi_{\min}}{2 \|\nabla_{\mathbf{u}}^* \psi(\mathbf{u}^{(k)})\|^2}.\quad (5.15)$$

■

If a priori we have knowledge of the value of ψ_{\min} , like in the case when there is no noise and the sources are identically distributed, the exact value of $\Delta^{(k)}$ can be computed and substituted to determine the NR step size (5.11). Otherwise, we propose to choose a coarse initial guess $\hat{\Delta}^{(k)}$, with progressive annealing to zero with the run of the iterations.

This should be done in such a way that the extraction vector could initially perform a fast descent in the contrast function, crossing back and forth the non-differentiable hyperplanes, until it is able to center at one of the local solutions. Once there, the annealing of the parameter will make the extraction vector to converge, bouncing around the solution with a progressively smaller deviation.

In the Figure 5.2, a geometrical interpretation of this step size is shown. The gradient $\nabla_{\mathbf{u}}^* \psi(\mathbf{u})$ yields to a fast descent based upon a linear approximation of the slope of the cost function.

5.3.2 A step size that uses the second-order information

We can derive an even faster (in the sense of the number of iterations) step size using the same Taylor expansion of the cost function, but now up to the second order by introducing the Hessian. We omit the iteration index k in some terms to simplify the expressions.

Theorem 5.2 (A step size based upon second-order Information)

The step size for the gradient descent algorithm based upon second-order information is given by

$$\mu_2^{(k)} = \frac{\|\underline{\nabla}_{\mathbf{u}} \psi\|^2 - \sqrt{\|\underline{\nabla}_{\mathbf{u}} \psi\|^4 - 2(\underline{\nabla}_{\mathbf{u}} \psi)^H \underline{\mathbf{H}}_{\mathbf{uu}} \psi (\underline{\nabla}_{\mathbf{u}} \psi) \Delta^{(k)}}}{(\underline{\nabla}_{\mathbf{u}} \psi)^H \underline{\mathbf{H}}_{\mathbf{uu}} \psi (\underline{\nabla}_{\mathbf{u}} \psi)}, \quad (5.16)$$

Where $\underline{\mathbf{H}}_{\mathbf{uu}} \psi$ is the augmented Hessian matrix of the cost function, defined by (5.7)-(5.9), and

$$\Delta^{(k)} = \psi(\mathbf{u}^{(k)}) - \psi_{\min} \quad (5.17)$$

is a nonnegative parameter which indicates how close is the algorithm to the convergence.

Proof: We follow the same procedure used in Theorem 5.1, but this time using a second-order approximation of (5.4),

$$\begin{aligned} \psi(\mathbf{u}^{(k+1)}) &= \psi(\mathbf{u}^{(k)}) + \underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u})(\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)}) \\ &\quad + \frac{1}{2}(\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)})^H \underline{\mathbf{H}}_{\mathbf{uu}} \psi(\mathbf{u})(\underline{\mathbf{u}}^{(k+1)} - \underline{\mathbf{u}}^{(k)}). \end{aligned} \quad (5.18)$$

By setting $\psi(\mathbf{u}^{(k+1)}) = \psi_{\min}$ in the left side of (5.18), and the update equation (5.13) on the right side, we obtain a quadratic equation,

$$\begin{aligned} \psi_{\min} &= \psi(\mathbf{u}^{(k)}) - \mu^{(k)} \|\underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u})\|^2 \\ &\quad + \frac{1}{2}(\mu^{(k)})^2 (\underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u}))^H \underline{\mathbf{H}}_{\mathbf{uu}} \psi(\mathbf{u})(\underline{\nabla}_{\mathbf{u}} \psi(\mathbf{u})). \end{aligned} \quad (5.19)$$

Whose terms can be ordered in the following way

$$\left(\frac{1}{2} (\underline{\nabla}_{\mathbf{u}} \psi)^H \underline{\mathbf{H}}_{\mathbf{uu}} \psi (\underline{\nabla}_{\mathbf{u}} \psi) \right) (\mu^{(k)})^2 - \|\underline{\nabla}_{\mathbf{u}} \psi\|^2 \mu^{(k)} + (\psi - \psi_{\min}) = 0. \quad (5.20)$$

Solving the quadratic equation and taking the lower value of the pair of solutions, we obtain the expression (5.16). ■

The Hessians defined in (5.8) and (5.9) need to be computed from the well-known gradient studied in Theorem 4.8. We exploit the structure introduced in (4.79), and the notation $\nabla_{\mathbf{u}^*}(\cdot) = \partial(\cdot) / \partial \mathbf{u}^*$, presented in Chapter 2.

$$\frac{\partial \psi}{\partial \mathbf{u}^*} = \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{L(\mathbf{y})}{\|\mathbf{u}\|} \right) = \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T \frac{1}{\|\mathbf{u}\|} + L(\mathbf{y}) \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T \in \mathbb{C}^n. \quad (5.21)$$

To obtain practical expressions for the Hessians, we use the following gradients:

$$\frac{\partial |\Delta y_i|}{\partial \mathbf{u}} = \frac{\partial (\Delta y_i \Delta y_i^*)^{1/2}}{\partial \mathbf{u}} = \frac{\Delta y_i \Delta \mathbf{z}_i^*}{2 |\Delta y_i|} \in \mathbb{C}^n, \quad (5.22)$$

$$\frac{\partial |\Delta y_i|}{\partial \mathbf{u}^*} = \frac{\partial (\Delta y_i \Delta y_i^*)^{1/2}}{\partial \mathbf{u}^*} = \frac{\Delta y_i^* \Delta \mathbf{z}_i}{2 |\Delta y_i|} \in \mathbb{C}^n. \quad (5.23)$$

$$\frac{\partial (\|\mathbf{u}\|^3)}{\partial \mathbf{u}} = \frac{3}{2} \|\mathbf{u}\| \mathbf{u}^* \in \mathbb{C}^n, \quad (5.24)$$

$$\frac{\partial (\|\mathbf{u}\|^3)}{\partial \mathbf{u}^*} = \frac{3}{2} \|\mathbf{u}\| \mathbf{u} \in \mathbb{C}^n. \quad (5.25)$$

$$\left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}} \right)^T = \sum_{i=1}^v \left(\frac{\partial |\Delta y_i|}{\partial \mathbf{u}} \right)^T = \sum_{i=1}^v \frac{\Delta y_i \Delta \mathbf{z}_i^H}{2 |\Delta y_i|} \in \mathbb{C}^n, \quad (5.26)$$

$$\left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T = \sum_{i=1}^v \left(\frac{\partial |\Delta y_i|}{\partial \mathbf{u}^*} \right)^T = \sum_{i=1}^v \frac{\Delta y_i^* \Delta \mathbf{z}_i^T}{2 |\Delta y_i|} \in \mathbb{C}^n, \quad (5.27)$$

$$\left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}} \right)^T = \frac{-\mathbf{u}^H}{2 \|\mathbf{u}\|^3} \in \mathbb{C}^n, \quad (5.28)$$

$$\left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T = \frac{-\mathbf{u}^T}{2 \|\mathbf{u}\|^3} \in \mathbb{C}^n. \quad (5.29)$$

In addition, the following Hessians are necessary:

$$\frac{\partial}{\partial \mathbf{u}} \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T = \sum_{i=1}^V \frac{\partial}{\partial \mathbf{u}} \left(\frac{\Delta \mathbf{y}_i^* \Delta \mathbf{z}_i^T}{2|\Delta \mathbf{y}_i|} \right) = \sum_{i=1}^V \frac{\Delta \mathbf{z}_i^* \Delta \mathbf{z}_i^H}{4|\Delta \mathbf{y}_i|} \in \mathbb{C}^{n \times n}, \quad (5.30)$$

$$\frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T = \sum_{i=1}^V \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\Delta \mathbf{y}_i^* \Delta \mathbf{z}_i^T}{2|\Delta \mathbf{y}_i|} \right) = - \sum_{i=1}^V \frac{(\Delta \mathbf{y}_i)^2 \Delta \mathbf{z}_i \Delta \mathbf{z}_i^T}{4|\Delta \mathbf{y}_i|} \in \mathbb{C}^{n \times n}, \quad (5.31)$$

$$\frac{\partial}{\partial \mathbf{u}} \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T = \frac{\partial}{\partial \mathbf{u}} \left(\frac{-\mathbf{u}^T}{2\|\mathbf{u}\|^3} \right) = \frac{3\mathbf{u}^* \mathbf{u}^T}{4\|\mathbf{u}\|^5} - \frac{\mathbf{I}}{2\|\mathbf{u}\|^3} \in \mathbb{C}^{n \times n}, \quad (5.32)$$

$$\frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T = \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{-\mathbf{u}^T}{2\|\mathbf{u}\|^3} \right) = \frac{3\mathbf{u} \mathbf{u}^T}{4\|\mathbf{u}\|^5} \in \mathbb{C}^{n \times n}. \quad (5.33)$$

With all these terms, we can build the Hessians by applying the operators $\partial(\cdot)/\partial \mathbf{u}$ and $\partial(\cdot)/\partial \mathbf{u}^*$ to the transpose gradient of the cost function (5.21),

$$\begin{aligned} \mathbf{H}_{\mathbf{u}} \psi &= \frac{\partial}{\partial \mathbf{u}} \left(\frac{\psi}{\partial \mathbf{u}^*} \right)^T = \frac{1}{\|\mathbf{u}\|} \frac{\partial}{\partial \mathbf{u}} \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T + \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}} \right) \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T \\ &\quad + \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}} \right) \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T + L(\mathbf{y}) \frac{\partial}{\partial \mathbf{u}} \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T \in \mathbb{C}^{n \times n}, \end{aligned} \quad (5.34)$$

$$\begin{aligned} \tilde{\mathbf{H}}_{\mathbf{u}} \psi &= \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\psi}{\partial \mathbf{u}^*} \right)^T = \frac{1}{\|\mathbf{u}\|} \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T + \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right) \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T \\ &\quad + \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right) \left(\frac{\partial L(\mathbf{y})}{\partial \mathbf{u}^*} \right)^T + L(\mathbf{y}) \frac{\partial}{\partial \mathbf{u}^*} \left(\frac{\partial(1/\|\mathbf{u}\|)}{\partial \mathbf{u}^*} \right)^T \in \mathbb{C}^{n \times n}. \end{aligned} \quad (5.35)$$

All these quantities can be computed within each iteration, and applied to the step size to obtain a fast descent towards the local minimum.

In the Figure 5.3, the value of the quantity $\psi(\mathbf{u}^{(k)}) - \psi_{\min}$ in a computer simulation is shown for the two step sizes proposed in Theorem 5.1. Attending to the number of iterations needed, the Newton-Raphson step size is a fast option for the descent of the cost function, but in general it is not as fast as the step size that uses the second order information.

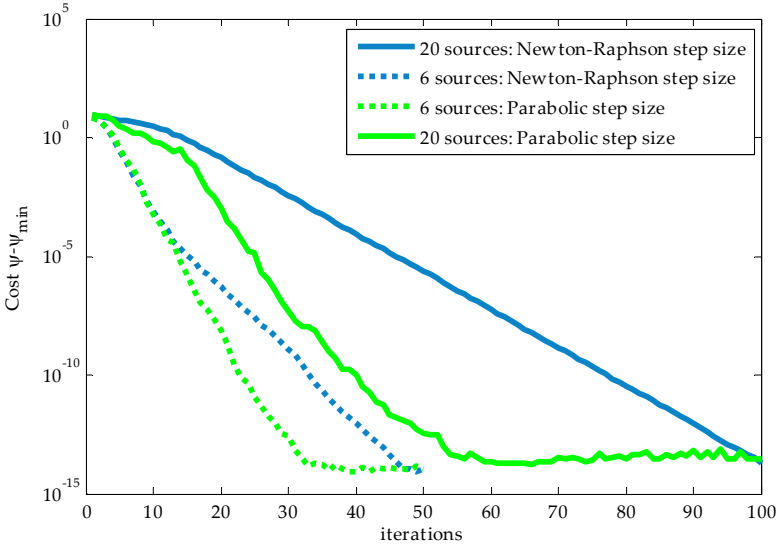


Figure 5.3 – This figure shows, in logarithmic scale, the comparison of the value of the distance to the convergence $\psi(\mathbf{u}^{(k)}) - \psi_{\min}$, when the Newton-Raphson step size (blue) and the step size that uses the second-order information (green) are used. We study two different cases: having 6 sources (dotted) and having 20 sources (solid). This plot was made assuming a noiseless situation, but the main effect of adding a random noise is the existence of a ground floor on the convergence curves¹⁵.

The mixing matrix has coefficients drawn from a Gaussian random process, and the number of iterations was set to 100. The length of the data was $T = 1000$ for the $n = m = 6$ case, and $T = 5000$ for the $n = m = 20$ case. It can be seen that the gap between the convergences of both step sizes lowers when the amount of the sources mixed is small. Indeed, for $n < 5$, the gain almost disappears.

It is important to remark that the computational complexity of the step size that uses the second-order information is greater than the linear approach, so the Newton-Raphson step size is usually the preferred option in low dimensionality problems.

The two step sizes proposed in this section are fast in practice, and can help to

¹⁵ In a noiseless situation, the value attained when the algorithm converges should be ideally zero. However, a value around 10^{-15} is attained due to the machine precision.

the method to converge in a few iterations. However, they do not guarantee the stability or the monotonous descent of the cost function within the iterations of the algorithm. This is a problem that we aim to address in the rest of the chapter.

5.4 A Step Size that Guarantees the Stability

In this Section, we want to propose a step size that guarantees the stability of the gradient descent algorithm. That is, our objective is to force conditions on the value of the step size, which may help to the algorithm to reach one of its local minima. The theory and the results obtained can be found in (Aguilera, 2013).

Let us define the convex perimeters of the sources as

$$L_i = L(s_i), i = 1, \dots, n. \quad (5.36)$$

The contrast function is invariant with respect to permutations in the order of the sources. Without loss of generality and for simplicity in the notation, we assume that the first source will be the one that the algorithm is locally extracting. Therefore, in the following we suppose $\psi_{\min} \equiv L_1$ (see the minimum reachable by (4.51)), which corresponds to the perimeter of the extracted source.

In spite that the algorithm with the Newton-Raphson or the step size that uses the second-order information has a fast local convergence, the global monotonous convergence to the extraction of one of the sources is not guaranteed. Additionally, as we will see later, this step size does not ensure that we extract the source that is closer in a certain sense to the algorithm initialization, which eventually difficults the determination of the correct ψ_{\min} to use in (5.10) or (5.16).

5.4.1 The underlying structure of the iteration

In this section, we analyze the structure of the gradient of the criterion in (5.1), because it is a key element in the determination of the convergence of the proposed algorithm. Let us group the convex perimeters of the sources in the vector

$$\mathbf{L}_S = \begin{bmatrix} L_1 \\ \vdots \\ L_n \end{bmatrix} \in \mathbb{R}^n, \quad (5.37)$$

and the moduli of the coefficients of the residual mixture vector

$$|\mathbf{g}| = \begin{bmatrix} |\mathcal{g}_1| \\ \vdots \\ |\mathcal{g}_n| \end{bmatrix} \in \mathbb{R}^n. \quad (5.38)$$

The following Theorem introduces the complex variable extension of the Euler homogeneous function Theorem for real variables (see (Apostol, 1967)).

Theorem 5.3 (Euler homogeneous function for real functions of complex variable)

For a real and *homogeneous function* $f(\mathbf{u}) : \mathbb{C}^n \rightarrow \mathbb{R}$ of degree $r \in \mathbb{R}$, i.e., when

$$f(\alpha \mathbf{u}) = \alpha^r f(\mathbf{u}), \quad \forall \alpha \in \mathbb{R}^+, \quad (5.39)$$

it holds that

$$rf(\mathbf{u}) = \langle \mathbf{u}, 2\nabla_{\mathbf{u}^*} f(\mathbf{u}) \rangle. \quad (5.40)$$

Proof: Let us define the complex vector $\mathbf{u}' \equiv \alpha \mathbf{u}$. Then,

$$\begin{aligned} r\alpha^{r-1}f(\mathbf{u}) &= \frac{\partial f}{\partial \alpha} = \sum_{i=1}^n \frac{\partial f}{\partial u'_i} \frac{\partial u'_i}{\partial \alpha} + \sum_{i=1}^n \frac{\partial f}{\partial (u'_i)^*} \frac{\partial (u'_i)^*}{\partial \alpha} \\ &= \sum_{i=1}^n \frac{\partial f}{\partial u'_i} u_i + \sum_{i=1}^n \frac{\partial f}{\partial (u'_i)^*} u_i^* = \sum_{i=1}^n \frac{\partial f}{\partial (\alpha u_i)} u_i + \sum_{i=1}^n \frac{\partial f}{\partial (\alpha u_i)^*} u_i^*. \end{aligned} \quad (5.41)$$

If $\alpha = 1$, then

$$rf(\mathbf{u}) = \sum_{i=1}^n \frac{\partial f}{\partial u_i} u_i + \sum_{i=1}^n \frac{\partial f}{\partial u_i^*} u_i^* = \langle \nabla_{\mathbf{u}} f(\mathbf{u}), \mathbf{u} \rangle + \langle \nabla_{\mathbf{u}^*} f(\mathbf{u}), \mathbf{u} \rangle, \quad (5.42)$$

which is twice $\langle \nabla_{\mathbf{u}^*} f(\mathbf{u}), \mathbf{u} \rangle$. ■

This is useful, for example, for the expression of the convex perimeter of the output as a scalar product:

$$L(\alpha y) = \alpha L(y), \quad \forall \alpha \neq 0 \Rightarrow L(y) = \langle \mathbf{u}, 2\nabla_{\mathbf{u}^*} L(y) \rangle. \quad (5.43)$$

A particularization of Theorem 5.3 leads to the following interesting result.

Theorem 5.4 (Orthogonality of the update equation)

Given the update equation (5.2), the extraction vector is orthogonal to its update at each iteration,

$$\left\langle \mathbf{u}^{(k)}, \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)} \right\rangle = 0. \quad (5.44)$$

And the same relation is also true for the global transfer vector,

$$\left\langle \mathbf{g}^{(k)}, \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} \right\rangle = 0. \quad (5.45)$$

Proof: The cost function $\psi(\mathbf{u}^{(k)})$ is scaling invariant (or homogeneous of degree $r = 0$), as stated in Section 4.6.1:

$$\psi(\alpha \mathbf{u}^{(k)}) = \psi(\mathbf{u}^{(k)}), \quad \forall \alpha \in \mathbb{R}^+. \quad (5.46)$$

Thus, Euler homogeneous function Theorem (5.40) for $r = 0$, states that the extraction vector is orthogonal to the gradient, and thus, to the update:

$$2 \left\langle \mathbf{u}^{(k)}, \nabla_{\mathbf{u}} \psi(\mathbf{u}^{(k)}) \right\rangle = r \psi(\mathbf{u}^{(k)}) = 0. \quad (5.47)$$

As $\mathbf{g}^{(k)} = \mathbf{V} \mathbf{u}^{(k)}$, the orthogonality is also true for the global transfer vector. ■

With these results, we are able to present the main result of this section in the following Theorem. It provides a geometrical interpretation of the update.

Theorem 5.5 (Geometrical interpretation of the update)

The iteration of the gradient descent BCA algorithm proposed in (5.2) can be rewritten, in terms of the moduli of the residual mixing coefficients, as

$$\left| \mathbf{g}^{(k+1)} \right| = \left| \mathbf{g}^{(k)} \right| - \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \left(\mathbf{I} - \frac{\left| \mathbf{g}^{(k)} \right| \left| \mathbf{g}^{(k)} \right|^{H}}{\left\| \mathbf{g}^{(k)} \right\|^2} \right) \mathbf{L}_s. \quad (5.48)$$

So the convergence of the iteration is only determined by three vectors of parameters: the initial residual mixture vector $\mathbf{g}^{(0)}$, the sequence of step sizes $\{\mu^{(k)} : k = 0, \dots, \infty\}$, and the perimeters of the sources aggregated in \mathbf{L}_s .

Proof: By using (5.43), the gradient of the cost function derived in (5.3) can

be expressed as

$$\nabla_{\mathbf{u}} \psi(\mathbf{u}) = \frac{1}{\|\mathbf{u}\|} \left(\nabla_{\mathbf{u}^*} L(y) - \frac{\mathbf{u}\mathbf{u}^H}{\|\mathbf{u}\|^2} \nabla_{\mathbf{u}^*} L(y) \right) = \left(\mathbf{I} - \frac{\mathbf{u}\mathbf{u}^H}{\|\mathbf{u}\|^2} \right) \frac{\nabla_{\mathbf{u}^*} L(y)}{\|\mathbf{u}\|}. \quad (5.49)$$

This implies that the gradient of the contrast function is proportional to the orthogonal projection of the gradient of the perimeter onto the *complementary subspace* of the subspace spanned by the extraction vector.

The extraction vector \mathbf{u} can be seen as a function of the residual mixing vector \mathbf{g} . As the unitary mixing matrix satisfy $\mathbf{V}^H \mathbf{V} = \mathbf{V} \mathbf{V}^H = \mathbf{I}$, we have $\|\mathbf{u}\| = \|\mathbf{g}\|$ and,

$$\mathbf{g}^H = \mathbf{u}^H \mathbf{V} \Leftrightarrow \mathbf{u}^H = \mathbf{g}^H \mathbf{V}^H. \quad (5.50)$$

With the help of the chain rule, we obtain

$$\nabla_{\mathbf{g}^*} \psi = \frac{\partial \psi}{\partial \mathbf{g}^*} = \left(\frac{\partial \mathbf{u}^*}{\partial \mathbf{g}^*} \right) \frac{\partial \psi}{\partial \mathbf{u}^*} = \mathbf{V}^H \nabla_{\mathbf{u}^*} \psi \in \mathbb{C}^n, \quad (5.51)$$

and also

$$\nabla_{\mathbf{u}^*} L(y) = \frac{\partial L(y)}{\partial \mathbf{u}^*} = \left(\frac{\partial \mathbf{g}^*}{\partial \mathbf{u}^*} \right) \frac{\partial L(y)}{\partial \mathbf{g}^*} = \mathbf{V} \nabla_{\mathbf{g}^*} L(y) \in \mathbb{C}^m. \quad (5.52)$$

By pre-multiplying the iteration (5.2) by \mathbf{V}^H , we obtain

$$\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} - \mu^{(k)} \nabla_{\mathbf{g}^*} \psi(\mathbf{u}), \quad (5.53)$$

where

$$\begin{aligned} \nabla_{\mathbf{g}^*} \psi(\mathbf{u}) &= \mathbf{V}^H \nabla_{\mathbf{u}^*} \psi(\mathbf{u}) = \mathbf{V}^H \left(\mathbf{I} - \frac{\mathbf{u}^{(k)} (\mathbf{u}^{(k)})^H}{\|\mathbf{g}^{(k)}\|^2} \right) \frac{\mathbf{V} \nabla_{\mathbf{g}^*} L(y)}{\|\mathbf{g}^{(k)}\|} \\ &= \left(\mathbf{I} - \frac{\mathbf{g}^{(k)} (\mathbf{g}^{(k)})^H}{\|\mathbf{g}^{(k)}\|^2} \right) \frac{\nabla_{\mathbf{g}^*} L(y)}{\|\mathbf{g}^{(k)}\|}. \end{aligned} \quad (5.54)$$

Note that this update (5.53) is expressed only in term of the residual mixing vector, which makes much easier the convergence analysis.

In order to be able to continue with the analysis of

$$\mathbf{g}^{(k+1)} = \mathbf{g}^{(k)} - \mu^{(k)} \left(\mathbf{I} - \frac{\mathbf{g}^{(k)} (\mathbf{g}^{(k)})^H}{\|\mathbf{g}^{(k)}\|^2} \right) \frac{\nabla_{\mathbf{g}^*} L(\mathbf{y})}{\|\mathbf{g}^{(k)}\|}, \quad (5.55)$$

we need to determine the hidden structure of $\nabla_{\mathbf{g}^*} L(\mathbf{y})$. As was shown in the previous Chapter, under the BCA hypothesis, the convex perimeter of $y = \mathbf{g}^H \mathbf{s}$ can be decomposed as a linear combination of the perimeter of the sources

$$L(\mathbf{y}) = \langle |\mathbf{g}|, \mathbf{L}_s \rangle = (|\mathbf{g}|)^H \mathbf{L}_s = \sum_{i=1}^n |g_i| L_i = \sum_{i=1}^n (g_i g_i^*)^{1/2} L_i. \quad (5.56)$$

This decomposition of the perimeter of the output is necessary for revealing the structure of its gradient with respect to the extraction vector. By recalling that

$$\frac{\partial |g_i|}{\partial g_i^*} = \frac{g_i}{2|g_i|} = \frac{1}{2} e^{j\angle g_i}, \quad \forall |g_i| > 0, \quad (5.57)$$

we obtain the gradient of the perimeter of the output,

$$\nabla_{\mathbf{g}^*} L(\mathbf{y}) = \left[\frac{\partial |g_i| L_i}{\partial g_i^*} \right]_i = \left(\frac{\partial |\mathbf{g}|}{\partial \mathbf{g}^*} \right) \mathbf{L}_s = \frac{1}{2} \mathbf{\Lambda} \mathbf{L}_s \in \mathbb{C}^n, \quad (5.58)$$

in terms of the vector of perimeters \mathbf{L}_s and the diagonal matrix unitary matrix $\mathbf{\Lambda} = (\mathbf{\Lambda}^H)^{-1}$, which is formed by the complex signs of the elements of the residual mixing vector,

$$\mathbf{\Lambda} = \begin{bmatrix} e^{j\angle g_1} & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{j\angle g_n} \end{bmatrix} \in \mathbb{C}^{n \times n}. \quad (5.59)$$

By decomposing the vector \mathbf{g} in phase and modulus contributions as $\mathbf{\Lambda} |\mathbf{g}|$, and by substituting (5.58) in (5.55), we can rewrite the iteration (5.55) as

$$\begin{aligned}
\mathbf{g}^{(k+1)} &= \Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| - \mu^{(k)} \left(\mathbf{I} - \frac{\Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| \left(\Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| \right)^H}{\left\| \mathbf{g}^{(k)} \right\|^2} \right) \frac{\Lambda^{(k)} \mathbf{L}_S}{2 \left\| \mathbf{g}^{(k)} \right\|} \\
&= \Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| - \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \left(\Lambda^{(k)} - \frac{\Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| \left(\left| \mathbf{g}^{(k)} \right| \right)^H \left(\Lambda^{(k)} \right)^H \Lambda^{(k)}}{\left\| \mathbf{g}^{(k)} \right\|^2} \right) \mathbf{L}_S \quad (5.60) \\
&= \Lambda^{(k)} \left(\left| \mathbf{g}^{(k)} \right| - \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \left(\mathbf{I} - \frac{\left| \mathbf{g}^{(k)} \right| \left(\left| \mathbf{g}^{(k)} \right| \right)^H}{\left\| \mathbf{g}^{(k)} \right\|^2} \right) \mathbf{L}_S \right).
\end{aligned}$$

Then, the proof of this Theorem is finished by taking modulus at both sides of this equation, and observing that its left-hand-side also decomposes as

$$\mathbf{g}^{(k+1)} = \Lambda^{(k+1)} \left| \mathbf{g}^{(k+1)} \right|. \quad (5.61)$$

■

Theorem 5.5 also provides a clear geometrical interpretation of this gradient descent iteration that is illustrated in Figure 5.4. The modulus of the residual mixture vector $\left| \mathbf{g}^{(k)} \right|$ is updated by subtracting to its value a term proportional to the orthogonal projection of the vector \mathbf{L}_S onto the subspace orthogonal to $\left| \mathbf{g}^{(k)} \right|$. In the figure, the following *orthogonal projector* is used:

$$P_{\left| \mathbf{g}^{(k)} \right|}^\perp = \mathbf{I} - \frac{\left| \mathbf{g}^{(k)} \right| \left(\left| \mathbf{g}^{(k)} \right| \right)^H}{\left\| \mathbf{g}^{(k)} \right\|^2}. \quad (5.62)$$

Thus, the iteration always enforces at these moduli to move apart from the vector of perimeters \mathbf{L}_S . As the vector $\left| \mathbf{g}^{(k)} \right|$ is constrained to the positive orthant, the iteration pushes its value toward the extraction solutions, which are the minima of the contrast function.

On one hand, when the argument inside the modulus of the right-hand-side of the iteration (5.48) is positive, the update of $\left| \mathbf{g}^{(k)} \right|$ is orthogonal to its current value. On the other hand, when this argument is negative, the orthogonality property is lost because the result after the orthogonal update is the reflected by the modulus operator.

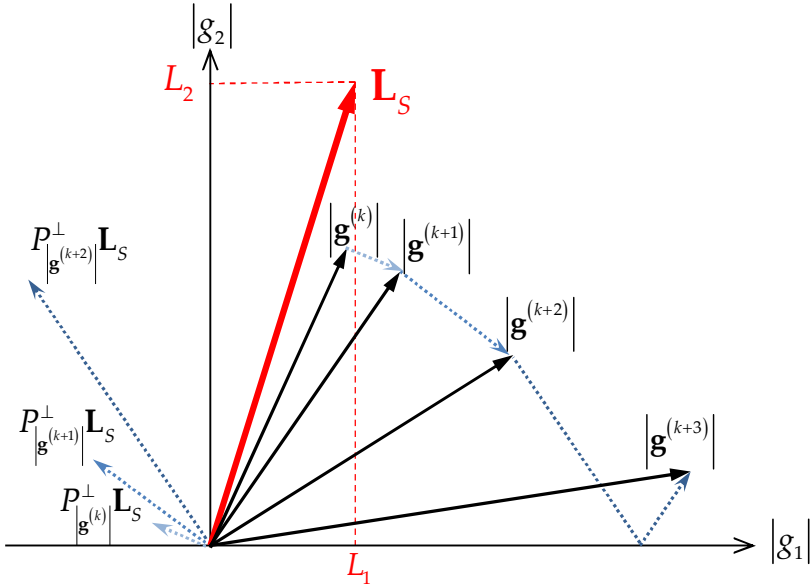


Figure 5.4 – This figure serves to illustrate the underlying geometrical interpretation of the BCA descent iteration, which has been addressed in Theorem 5.5. As we are in the two-dimensional plane, we presume a mixture of two sources ($n = 2$) with perimeters L_1 and L_2 . In the figure, the notation $P_{|\mathbf{g}^{(i)}}^\perp$ has been used to abbreviate the orthogonal projector.

5.4.2 A convenient parameterization

Before being able to start with the convergence study, we need first to find a convenient parameterization that decouples, as much as possible, the cross-dependence on the elements of the vector of parameters at two consecutive iterations of the algorithm given by (5.48). This is achieved by the transformation of the parameters that is presented below.

Definition (Non-negative normalized coefficients): Consider the parameterization

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} \in \mathbb{R}^n \quad (5.63)$$

of the residual mixing vector \mathbf{g} . Its coefficients are determined by the one-to-one transformation:

$$\beta_i^{(k)} = \frac{|\mathbf{g}_i^{(k)}|}{\|\mathbf{g}\| L_i}, \quad i = 1, \dots, n. \quad (5.64)$$

Under this parameterization, the BCA iteration (5.48) simplifies to the following Theorem.

Theorem 5.6 (The iteration parameterized)

The update equation (5.48) can be written in terms of the non-negative normalized coefficients:

$$\boldsymbol{\beta}^{(k+1)} = \alpha^{(k)} \left| \boldsymbol{\beta}^{(k)} - \gamma^{(k)} \mathbf{1} \right| \in \mathbb{C}^n, \quad (5.65)$$

where $\mathbf{1}$ is a column vector of n ones, the scalar term $\alpha^{(k)}$ is defined by

$$\alpha^{(k)} = \left(2 \|\mathbf{g}^{(k)}\| \|\mathbf{g}^{(k+1)}\| \frac{\mu^{(k)}}{\gamma^{(k)}} \right)^{-1} \geq 0, \quad (5.66)$$

and $\gamma^{(k)}$ is the *normalized step size*, given by

$$\gamma^{(k)} = \left(\frac{2 \|\mathbf{g}^{(k)}\|^2}{\mu^{(k)}} + \psi(\mathbf{u}^{(k)}) \right)^{-1} \geq 0. \quad (5.67)$$

Proof: The $\beta_i^{(k)}$ coefficients can be rearranged in the column vector $\boldsymbol{\beta}^{(k)}$:

$$\boldsymbol{\beta}^{(k)} = \frac{\text{diag}^{-1}(\mathbf{L}_S) \left| \mathbf{g}^{(k)} \right|}{\|\mathbf{g}^{(k)}\|}. \quad (5.68)$$

From (5.56), we express the cost function as

$$\psi(\mathbf{u}^{(k)}) = \frac{L(\mathbf{y})}{\|\mathbf{u}^{(k)}\|} = \frac{\left(\left| \mathbf{g}^{(k)} \right| \right)^H \mathbf{L}_S}{\|\mathbf{g}^{(k)}\|}. \quad (5.69)$$

On the other hand, by substituting (5.69) into the last row of (5.60), we can observe three terms:

$$\mathbf{g}^{(k+1)} = \Lambda^{(k)} \left| \mathbf{g}^{(k)} \right| - \Lambda^{(k)} \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \mathbf{L}_s + \Lambda^{(k)} \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|^2} \left| \mathbf{g}^{(k)} \right| \psi \left(\mathbf{u}^{(k)} \right). \quad (5.70)$$

By substituting $\left| \mathbf{g}^{(k)} \right| = \text{diag}(\mathbf{L}_s) \beta^{(k)} \left\| \mathbf{g}^{(k)} \right\|$ into (5.70),

$$\begin{aligned} \mathbf{g}^{(k+1)} &= \Lambda^{(k)} \left\| \mathbf{g}^{(k)} \right\| \text{diag}(\mathbf{L}_s) \beta^{(k)} - \Lambda^{(k)} \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \mathbf{L}_s \\ &\quad + \Lambda^{(k)} \frac{\mu^{(k)} \psi \left(\mathbf{u}^{(k)} \right)}{2 \left\| \mathbf{g}^{(k)} \right\|} \text{diag}(\mathbf{L}_s) \beta^{(k)}. \end{aligned} \quad (5.71)$$

The modulus of the expression above can be compacted element-wise as

$$\left| g_i^{(k+1)} \right| = \left| L_i \beta_i^{(k)} \left(\left\| \mathbf{g}^{(k)} \right\| + \frac{\mu^{(k)} \psi \left(\mathbf{u}^{(k)} \right)}{2 \left\| \mathbf{g}^{(k)} \right\|} \right) - \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} L_i \right|, \quad (5.72)$$

which leads to the following relation between $\beta_i^{(k)}$ and $\beta_i^{(k+1)}$ when is substituted into the definition (5.64) for the $(k+1)$ -th iteration:

$$\beta_i^{(k+1)} = \frac{\left| \beta_i^{(k)} \left(\left\| \mathbf{g}^{(k)} \right\| + \frac{\mu^{(k)} \psi \left(\mathbf{u}^{(k)} \right)}{2 \left\| \mathbf{g}^{(k)} \right\|} \right) - \frac{\mu^{(k)}}{2 \left\| \mathbf{g}^{(k)} \right\|} \right|}{\left\| \mathbf{g}^{(k+1)} \right\|}. \quad (5.73)$$

Finally, if we take out common factor $\left(2 \left\| \mathbf{g}^{(k)} \right\| \right)^{-1}$ in the numerator, the expression (5.73) simplifies to

$$\beta_i^{(k+1)} = \frac{\left| \beta_i^{(k)} \left(2 \left\| \mathbf{g}^{(k)} \right\|^2 + \mu^{(k)} \psi \left(\mathbf{u}^{(k)} \right) \right) - \mu^{(k)} \right|}{2 \left\| \mathbf{g}^{(k)} \right\| \left\| \mathbf{g}^{(k+1)} \right\|} \quad (5.74)$$

which is the same result of the Theorem, given by (5.65)-(5.67). ■

This Theorem tells us that between two consecutive iterations, the beta parameters undergo a common subtraction of $\gamma^{(k)}$, a modulus operation, and a common scaling by $\alpha^{(k)}$.

In this way, the update (5.65) reveals that there is a specific dependence of each coefficient $\beta_i^{(k)}$ on the value of the same coefficient at the next iteration $\beta_i^{(k+1)}$ and a common shared interaction with the remaining elements $\beta_j^{(k)}$, with $j \neq i$, through the scalars $\alpha^{(k)}$ and $\gamma^{(k)}$.

The global monotonous convergence condition

We introduce a permutation π of the set of indexes $\{1, \dots, n\}$ that rearranges them in descending order. At k -th iteration, the $\beta_{\pi(i)}^{(k)}$ coefficients are sorted such as:

$$\beta_{\pi(1)}^{(k)} \geq \beta_{\pi(2)}^{(k)} \geq \dots \geq \beta_{\pi(n)}^{(k)}. \quad (5.75)$$

After each update (5.65), the new coefficients do not satisfy the previous ordering, but the corresponding permutation π rearranges them again in descending order

$$\beta_{\pi(1)}^{(k+1)} \geq \beta_{\pi(2)}^{(k+1)} \geq \dots \geq \beta_{\pi(n)}^{(k+1)}. \quad (5.76)$$

This is illustrated in the following example.

Example 5.1 Suppose $n=4$ sources. On a given iteration k , the coefficients $\beta_{\pi(i)}^{(k)}$ are sorted always as

$$\beta_{\pi(1)}^{(k)} \geq \beta_{\pi(2)}^{(k)} \geq \dots \geq \beta_{\pi(n)}^{(k)}, \quad (5.77)$$

which, e.g., corresponds to

$$\frac{|\mathcal{G}_2^{(k)}|}{\|\mathbf{g}\|L_2} \geq \frac{|\mathcal{G}_3^{(k)}|}{\|\mathbf{g}\|L_3} \geq \frac{|\mathcal{G}_1^{(k)}|}{\|\mathbf{g}\|L_1} \geq \frac{|\mathcal{G}_4^{(k)}|}{\|\mathbf{g}\|L_4}, \quad (5.78)$$

so $\beta_2^{(k+1)} \geq \beta_3^{(k+1)} \geq \beta_1^{(k+1)} \geq \beta_4^{(k+1)}$ and $\pi^{(k)} = [2 \ 3 \ 1 \ 4]$.

At the next iteration $k+1$, the coefficients $|\mathcal{G}_i^{(k+1)}|$, $i=1, \dots, 4$ changes such that

$$\frac{|\mathcal{G}_2^{(k+1)}|}{\|\mathbf{g}\|L_2} \geq \frac{|\mathcal{G}_1^{(k+1)}|}{\|\mathbf{g}\|L_1} \geq \frac{|\mathcal{G}_3^{(k+1)}|}{\|\mathbf{g}\|L_3} \geq \frac{|\mathcal{G}_4^{(k+1)}|}{\|\mathbf{g}\|L_4}. \quad (5.79)$$

The ordering of the $\beta_{\pi(i)}^{(k+1)}$ coefficients is always $\beta_{\pi(1)}^{(k+1)} \geq \beta_{\pi(2)}^{(k+1)} \geq \beta_{\pi(3)}^{(k+1)} \geq \beta_{\pi(4)}^{(k+1)}$.

By comparing (5.78) and (5.79), one can see that the second and third coefficients are permuted at the execution of this iteration,

$$\beta_2^{(k+1)} \geq \beta_1^{(k+1)} \geq \beta_3^{(k+1)} \geq \beta_4^{(k+1)}, \quad (5.80)$$

so to the reordering at this update is $\pi^{(k+1)} = [2 \ 1 \ 3 \ 4]$.

The following definition establishes the criterion for monotonous convergence in terms of the parameterization.

Definition (The global monotonous convergence condition): In order to guarantee a monotonous descent of all the quotients $\beta_{\pi(i)} / \beta_{\pi(1)}$, the following condition must be satisfied,

$$c_i^{(k+1)} < c_2^{(k)}, \text{ for } i > 2, \quad (5.81)$$

where the *convergence quotients* c_i has been introduced to ease the notation:

$$c_i^{(k)} = \beta_{\pi(i)}^{(k)} / \beta_{\pi(1)}^{(k)}. \quad (5.82)$$

That is, at each iteration, the second peak c_2 of the quotients spectrum is greater than the remaining quotients c_i , $i > 2$, at the next iteration. As the coefficients can permute their positions through the iterations, this guarantees that the second peak decreases continuously.

5.4.3 Stable bounds for the step size

In this section, we derive three upper bounds for the step size so as to guarantee different desired behaviors of the descent algorithm. In Figure 5.5, we illustrate the reasoning made for the derivation of the step sizes. For the sake of simplicity, $n = 3$ is supposed and $\pi = [1 \ 2 \ 3]$, so we have big chances of extracting the first source. To analyze the worst case, β_3 is zero at the current iteration ($|\beta_3 - \gamma|$ may grow rapidly if the step size is too large).

The letters without mark correspond to the points (β_2, β_1) , and the letters with mark (') correspond to the points (β_3, β_1) . It is important to remark that after the update of the β_i coefficients, there is a rearrangement (that is not shown in the figure) to sort them in descending value.

We analyze six different situations of the points (β_i, β_1) , depending on the value of the step size:

- The initial state of the system is given by the points O and O' .
- When the step size is small, the state of the system move to the position located between the points shown by O and O' , and the points shown by P and P' . The coefficient $\pi(2)$ does not change ($\pi(2) = 2$).

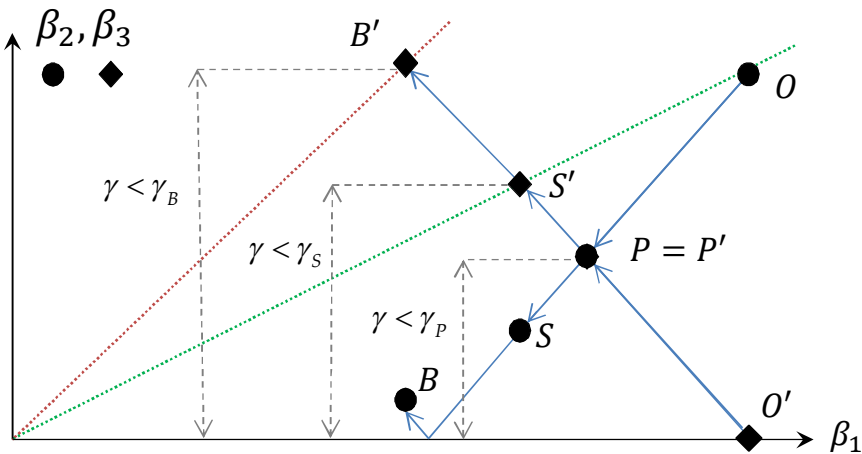


Figure 5.5 – Geometrical interpretation of the movement of the points (β_i, β_1) , for $i = 2, 3$, depending on the value of the step size. The blue arrows show the movement of the coefficients during two consecutive iterations. The black circles O, P, S , and B corresponds to (β_2, β_1) , while the black squares O', P', S' , and B' corresponds to (β_3, β_1) , at any given iteration. The green dotted line shows the limit for the global monotonous convergence (in the sense of the descent of $c_2^{(k)}$), which the moving points cannot exceed to satisfy the condition (5.81). The red dotted line shows the limit for the prevention of a leakage of the iteration to another basin of the cost function.

- Eventually, the step size grows and they both reach the point $P = P'$, where the coefficients satisfy $\beta_2 = \beta_3$. The step size derived, μ_p , is an upper bound to prevent the change of $\pi(2)$.
- Hereinafter, the coefficient $\pi(2)$ may change to $\pi(2) = 3$ if the step size gets larger. When this happens, the iteration is still monotonous convergent until the points S and S' are reached. This latter state is given by the intersection of the green dotted line (determined by the initial black circle) and the blue arrows. At this situation, we can derive the bound for the step size μ_s , which guarantees the stability of the algorithm.
- If the step size grows from this value, we cannot guarantee the monotonous descent of the quotient c_2 , which is our condition for the stability of the algorithm. This situation is shown by any position between the points S and S' , and the points B and B' , where the condition (5.81) is violated.

- Finally, if the step size gets sufficiently large, the points B and B' are reached. This latter state is given by the intersection of the red dotted line (given by the locus of points where $\beta_{\pi(2)} = \beta_{\pi(1)}$) and the blue arrows. There is a risk of change of the $\pi(1)$ coefficient which potentially changes the basin of the cost function where the iteration is located. The step size μ_B , derived from this condition, is an upper bound to *prevent the leakage of the algorithm from the initial basin of the cost function*.

The following subsections detail the derivation of these three upper bounds, using the normalized step size γ .

A step size that prevents the change of the $\pi(2)$ coefficient: μ_P

The following Theorem establishes this initial upper bound for the step size, which can be seen in Figure 5.5 for its normalized version γ_P .

Theorem 5.7 (A step size that prevents the change of the $\pi(2)$ coefficient)

Given the extraction vector $\mathbf{u}^{(k)}$, the value of the cost function $\psi(\mathbf{u}^{(k)})$, and the coefficient $\beta_{\pi(2)}^{(k)}$, any step size $\mu^{(k)} \leq \mu_P^{(k)}$ prevents the change of the $\pi(2)$ coefficient, where

$$\mu_P^{(k)} = \frac{2 \|\mathbf{u}^{(k)}\|^2}{\frac{2}{\beta_{\pi(2)}^{(k)}} - \psi(\mathbf{u}^{(k)})}. \quad (5.83)$$

Proof: Let us suppose the worst case, where

$$\beta_{\pi(i)}^{(k)} \approx 0, \text{ for } i \geq 3. \quad (5.84)$$

This means that, when the iteration (5.65) executes, the coefficients $\beta_{\pi(i)}^{(k)}$, $i \geq 3$ are going to rise up due to the absolute value, and so are in risk of becoming greater than $\beta_{\pi(2)}^{(k)}$ (there is a change of the $\pi(2)$ coefficient).

We can impose a condition on $\gamma^{(k)}$, so (5.81) is satisfied:

$$c_i^{(k+1)} = \frac{\beta_{\pi(i)}^{(k+1)}}{\beta_{\pi(1)}^{(k+1)}} = \frac{|\beta_{\pi(i)}^{(k)} - \gamma^{(k)}|}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} \leq \frac{\beta_{\pi(2)}^{(k)} - \gamma^{(k)}}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} < \frac{\beta_{\pi(2)}^{(k)}}{\beta_{\pi(1)}^{(k)}} = c_2^{(k)}, \text{ for } i \geq 2. \quad (5.85)$$

The absence of the modulus operator in the denominator of the second equality is due to the fact that $\beta_{\pi(1)}^{(k)} > \gamma^{(k)}$, which imposes an initial upper bound for the normalized step size. However, by noting (5.84), the middle inequality imposes an stricter condition on $\gamma^{(k)}$:

$$\left| \beta_{\pi(i)}^{(k)} - \gamma^{(k)} \right| < \gamma^{(k)} \leq \beta_{\pi(2)}^{(k)} - \gamma^{(k)}, \text{ for } i \geq 3. \quad (5.86)$$

Finally, the last inequality is due to the fact that we are subtracting the same quantity $\gamma^{(k)}$ to both numerator and denominator.

If the condition (5.86) is satisfied, the coefficient $\pi(2)$ does not change between two consecutive iterations. By noting that $\|\mathbf{u}^{(k)}\| = \|\mathbf{g}^{(k)}\|$ and by substituting (5.67) in (5.86), one can translate this condition from $\gamma^{(k)}$ to $\mu^{(k)}$,

$$\frac{2\mu^{(k)}}{2\|\mathbf{u}^{(k)}\|^2 + \mu^{(k)}\psi(\mathbf{u}^{(k)})} \leq \beta_{\pi(2)}^{(k)} \Rightarrow \mu^{(k)} \left(2 - \psi(\mathbf{u}^{(k)}) \beta_{\pi(2)}^{(k)} \right) \leq 2\|\mathbf{u}^{(k)}\|^2 \beta_{\pi(2)}^{(k)}. \quad (5.87)$$

Finally, by solving for $\mu^{(k)}$, we get the initial upper bound for the step size presented in this Theorem. ■

This step size prevents the change of the $\pi(2)$ coefficient, through iterations. It is an upper bound that guarantees the global monotonous convergence. However, it is slow when compared to Newton-Raphson or step size that uses the second order information, so another fast and stable upper bound can be proposed.

A step size that guarantees the monotonous convergence: μ_S

There is still a margin for $\mu^{(k)}$ to satisfy the monotonous convergence condition (5.81). Thus, a larger upper bound for the step size can be established. The next Theorem presents another upper bound for $\mu^{(k)}$, which can be seen in Figure 5.5 for its normalized version γ_S .

Theorem 5.8 (A step size that guarantees the monotonous convergence)

Given the extraction vector $\mathbf{u}^{(k)}$, the value of the cost function $\psi(\mathbf{u}^{(k)})$, and the coefficients $\beta_{-\langle i \rangle}^{(k)}$ and $\beta_{-\langle j \rangle}^{(k)}$, any step size $\mu^{(k)} < \mu_S^{(k)}$ guarantees the stability of the algorithm (in the sense of the monotonous descent of $c_2^{(k)}$), where

$$\mu_s^{(k)} = \frac{2 \|\mathbf{u}^{(k)}\|^2}{\frac{1}{\beta_{\pi(2)}^{(k)}} + \frac{1}{\beta_{\pi(1)}^{(k)}} - \psi(\mathbf{u}^{(k)})}. \quad (5.88)$$

Proof: Although the condition (5.86) is not satisfied now, we can derive an alternative condition for $\gamma^{(k)}$, so (5.81) holds:

$$c_i^{(k+1)} = \frac{\beta_{\pi(i)}^{(k+1)}}{\beta_{\pi(1)}^{(k+1)}} = \frac{|\beta_{\pi(i)}^{(k)} - \gamma^{(k)}|}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} \leq \frac{\gamma^{(k)}}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} < \frac{\beta_{\pi(2)}^{(k)}}{\beta_{\pi(1)}^{(k)}} = c_2^{(k)}, \text{ for } i \geq 2. \quad (5.89)$$

The second equality is due to the same fact that we exposed in (5.85). The middle inequality comes from applying (5.84) to the absolute value. The last inequality forces a condition on $\gamma^{(k)}$ so as to (5.81) is strictly satisfied:

$$\frac{\gamma^{(k)}}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} < \frac{\beta_{\pi(2)}^{(k)}}{\beta_{\pi(1)}^{(k)}} \Rightarrow \gamma^{(k)} < \beta_{\pi(2)}^{(k)} - \frac{\beta_{\pi(2)}^{(k)}}{\beta_{\pi(1)}^{(k)}} \gamma^{(k)} \quad (5.90)$$

In this case, condition (5.86) is not necessarily satisfied, so $\pi(2)$ may change. However, this is not a risk for the convergence, as long as (5.81) is still satisfied. Therefore, the quotient $c_2^{(k)}$ has a monotonous descent through the iterations.

By noting that $\|\mathbf{u}^{(k)}\| = \|\mathbf{g}^{(k)}\|$ and by substituting (5.67) in (5.86), one can translate this condition from $\gamma^{(k)}$ to $\mu^{(k)}$,

$$\frac{1 + \beta_{\pi(2)}^{(k)} / \beta_{\pi(1)}^{(k)}}{2 \frac{\|\mathbf{g}^{(k)}\|^2}{\mu^{(k)}} + \psi(\mathbf{u}^{(k)})} < \beta_{\pi(2)}^{(k)} \Rightarrow \mu^{(k)} \left(1 + \frac{\beta_{\pi(2)}^{(k)}}{\beta_{\pi(1)}^{(k)}} - \psi(\mathbf{u}^{(k)}) \right) < 2 \|\mathbf{u}^{(k)}\|^2 \beta_{\pi(2)}^{(k)}. \quad (5.91)$$

Finally, by solving for $\mu^{(k)}$, we get the upper bound for the step size presented in this Theorem. ■

The value $\mu_s^{(k)}$, which is greater than $\mu_p^{(k)}$, is the maximum value for the step size which guarantees that the global monotonous convergence condition is true.

A step size that prevents the leakage from the current basin of the cost function: μ_B

The next bound for the step size is located in the region where we cannot guarantee the monotonous convergence of the algorithm. However, it is important to know whether the update equation can lead the extraction vector to a point of the surface of the cost function that is outside of the basin where it was initialized.

If so, a convenient initialization may not guarantee the convergence to a certain solution, because it is possible for the iteration to escape from the desired basin. This can be seen in Figure 5.1, where the cost function has two basins, each one corresponding to the extraction of one of the two sources. The next Theorem presents another upper bound for $\mu^{(k)}$, which can be seen in Figure 5.5 for its normalized version γ_B .

Theorem 5.9 (A step size that prevents the leakage from the initial basin of the cost function)

Given the extraction vector $\mathbf{u}^{(k)}$, the value of the cost function $\psi(\mathbf{u}^{(k)})$, and the coefficient $\beta_{\pi(i)}^{(k)}$, any step size $\mu^{(k)} \leq \mu_B^{(k)}$ prevents the leakage of the algorithm from the basin where it is located, where

$$\mu_B^{(k)} = \frac{2 \|\mathbf{u}^{(k)}\|^2}{\frac{2}{\beta_{\pi(1)}^{(k)}} - \psi(\mathbf{u}^{(k)})}. \quad (5.92)$$

Proof: The step sizes $\mu^{(k)} > \mu_S^{(k)}$ do not guarantee the monotonous convergence of the algorithm (in the sense of the descent of $c_2^{(k)}$). However, a condition for the preservation of the actual basin is given by ensuring that

$$c_i^{(k+1)} = \frac{\beta_{\pi(i)}^{(k+1)}}{\beta_{\pi(1)}^{(k+1)}} = \frac{|\beta_{\pi(i)}^{(k)} - \gamma^{(k)}|}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} \leq \frac{\gamma^{(k)}}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} < \frac{\beta_{\pi(1)}^{(k)}}{\beta_{\pi(1)}^{(k)}} = 1, \quad \text{for } i \geq 2. \quad (5.93)$$

The process is the same that we followed in the proof of Theorem 5.8. Only the last inequality differs, and forces a condition on $\gamma^{(k)}$ so as to $c_i^{(k+1)} < 1$ is strictly satisfied:

$$\frac{\gamma^{(k)}}{\beta_{\pi(1)}^{(k)} - \gamma^{(k)}} < 1 \Rightarrow \gamma^{(k)} < \beta_{\pi(1)}^{(k)} - \gamma^{(k)} \quad (5.94)$$

By noting that $\|\mathbf{u}^{(k)}\| = \|\mathbf{g}^{(k)}\|$ and by substituting (5.67) in (5.94), one can translate again this condition from $\gamma^{(k)}$ to $\mu^{(k)}$,

$$\frac{2}{\beta_{\pi(1)}^{(k)}} < \frac{2\|\mathbf{g}^{(k)}\|^2}{\mu^{(k)}} + \psi(\mathbf{u}^{(k)}) \Rightarrow \left(\frac{2}{\beta_{\pi(1)}^{(k)}} - \psi(\mathbf{u}^{(k)}) \right) \mu^{(k)} < 2\|\mathbf{u}^{(k)}\|^2. \quad (5.95)$$

Finally, by solving for $\mu^{(k)}$, we get the main result of this Theorem. ■

The value $\mu_B^{(k)} > \mu_S^{(k)} > \mu_P^{(k)}$ is the maximum value for the step size that guarantees that the algorithm does not escapes from the basin where it is.

5.4.4 Practical estimators: the proposed step size

As we have said, the step sizes $\mu^{(k)} > \mu_S^{(k)}$ do not guarantee the stability of the algorithm, so one cannot use them in practice. Nevertheless, the upper bounds $\mu_S^{(k)}$ and $\mu_P^{(k)}$ presented in the previous discussion may be used, but they depend on unknown values. In this subsection, we provide practical estimators for them, so they can be computed and used in adaptive algorithms.

Evaluating the norms of the gradients

We can combine the decomposition of the gradient of the perimeter (5.58) with the fact that the matrix \mathbf{V} in (5.52) belongs to the unitary group, to express the norm of the vector of perimeters as

$$\|\mathbf{L}_S\| = 2\|\nabla_{\mathbf{g}} L(y)\| = 2\|\nabla_{\mathbf{u}} L(y)\|. \quad (5.96)$$

On one hand, the perimeter $L(y)$ can also be expressed in terms of the inner product (5.43), so the cost is

$$\psi(\mathbf{u}) = \frac{L(y)}{\|\mathbf{u}\|} = \frac{\langle 2\mathbf{u}, \nabla_{\mathbf{u}} L(y) \rangle}{\|\mathbf{u}\|}. \quad (5.97)$$

On the other hand, the norm of the gradient of the cost function is defined by the inner product

$$\|\nabla_{\mathbf{u}} \psi(\mathbf{u})\|^2 = \langle \nabla_{\mathbf{u}} \psi(\mathbf{u}), \nabla_{\mathbf{u}} \psi(\mathbf{u}) \rangle. \quad (5.98)$$

By substituting (5.3) in (5.98),

$$\begin{aligned} \|\nabla_{\mathbf{u}} \psi(\mathbf{u})\|^2 = & \left\langle \frac{\nabla_{\mathbf{u}} L(y)}{\|\mathbf{u}\|}, \frac{\nabla_{\mathbf{u}} L(y)}{\|\mathbf{u}\|} \right\rangle + \left\langle \frac{\mathbf{u}}{2\|\mathbf{u}\|^2} \psi(\mathbf{u}), \frac{\mathbf{u}}{2\|\mathbf{u}\|^2} \psi(\mathbf{u}) \right\rangle \\ & - 2 \left\langle \frac{\nabla_{\mathbf{u}} L(y)}{\|\mathbf{u}\|}, \frac{\mathbf{u}}{2\|\mathbf{u}\|^2} \psi(\mathbf{u}) \right\rangle. \end{aligned} \quad (5.99)$$

The first two terms are norms, and the third one can be simplified by using the relation (5.97), so

$$\|\nabla_{\mathbf{u}} \psi(\mathbf{u})\|^2 = \frac{\|\nabla_{\mathbf{u}} L(y)\|^2}{\|\mathbf{u}\|^2} + \frac{\|\mathbf{u}\|^2 \psi(\mathbf{u})^2}{4\|\mathbf{u}\|^4} - \frac{\langle 2\mathbf{u}, \nabla_{\mathbf{u}} L(y) \rangle}{2\|\mathbf{u}\|^3} \psi(\mathbf{u}) \quad (5.100)$$

Finally, by using the equivalence (5.96) in the first term of the right side,

$$\|\nabla_{\mathbf{u}} \psi(\mathbf{u})\|^2 = \frac{\|\mathbf{L}_s\|^2}{4\|\mathbf{u}\|^2} + \frac{\|\mathbf{u}\|^2 (\psi(\mathbf{u}))^2}{4\|\mathbf{u}\|^4} - \frac{(\psi(\mathbf{u}))^2}{2\|\mathbf{u}\|^2}, \quad (5.101)$$

which is equivalent to

$$\|\nabla_{\mathbf{u}} \psi(\mathbf{u})\|^2 = \frac{\|\mathbf{L}_s\|^2 - (\psi(\mathbf{u}))^2}{4\|\mathbf{u}\|^2}. \quad (5.102)$$

This relation will be used later to relate two interesting step sizes.

Bounds for the non-negative normalized coefficients

In the following Theorems, some useful bounds for the $\beta_{\pi(i)}^{(k)}$ parameters are presented. We use the “hat” notation $\hat{\beta}_{\pi(i)}^{(k)}$ because these bounds can also act as practical estimators, so they can be computed at each iteration from the available data.

Theorem 5.10 (Upper and lower bounds for the $\beta_{\pi(1)}$ coefficient)

The non-negative normalized coefficient $\beta_{\pi(1)}^{(k)}$ is upper and lower bounded:

$$\frac{1}{\hat{\beta}_{\pi(1),MIN}^{(k)}} \geq \frac{1}{\beta_{\pi(1)}^{(k)}} \geq \frac{1}{\hat{\beta}_{\pi(1),MAX}^{(k)}}, \quad (5.103)$$

with

$$\frac{1}{\hat{\beta}_{\pi(1),MAX}^{(k)}} \triangleq L_{\pi(1)}, \quad (5.104)$$

and

$$\frac{1}{\hat{\beta}_{\pi(1),MIN}^{(k)}} \triangleq 2 \left\| \mathbf{u}^{(k)} \right\|^2 \left(\frac{\psi(\mathbf{u}^{(k)}) - L_{\pi(1)}}{\psi(\mathbf{u}^{(k)})} \right) \left(\mu_{NR}^{(k)} \right)^{-1} + \psi(\mathbf{u}^{(k)}). \quad (5.105)$$

Where $\mu_{NR}^{(k)}$ is the Newton-Raphson step size from Theorem 5.1.

Proof: On one hand, the upper bound can be obtained from the definition (5.64). Since $\left| g_{\pi(1)}^{(k)} \right| \leq \left\| \mathbf{g}^{(k)} \right\|$, we have

$$\frac{1}{\beta_{\pi(1)}^{(k)}} \geq L_{\pi(1)}. \quad (5.106)$$

On the other hand, the lower bound for $\beta_{\pi(1)}^{(k)}$, namely $\hat{\beta}_{\pi(1),MIN}^{(k)}$, is not as tight as the previous one. By using (5.56) into the cost function (5.1),

$$\psi(\mathbf{u}^{(k)}) = \sum_{i=1}^n \beta_{\pi(i)}^{(k)} L_{\pi(i)}^2 = \beta_{\pi(1)}^{(k)} L_{\pi(1)}^2 + \sum_{i=2}^n \beta_{\pi(i)}^{(k)} L_{\pi(i)}^2. \quad (5.107)$$

By noting that $\beta_{\pi(1)}^{(k)} \geq \beta_{\pi(i)}^{(k)}$, for $i \geq 2$, we have

$$\psi(\mathbf{u}^{(k)}) \leq \beta_{\pi(1)}^{(k)} \left\| \mathbf{L}_S \right\|^2. \quad (5.108)$$

Therefore, by using the relation (5.102), the lower bound for $\beta_{\pi(1)}^{(k)}$ is given by

$$\frac{1}{\beta_{\pi(1)}^{(k)}} \leq \frac{4 \left\| \mathbf{u}^{(k)} \right\|^2 \left\| \nabla_{\mathbf{u}} \psi(\mathbf{u}^{(k)}) \right\|^2 + \psi(\mathbf{u}^{(k)})^2}{\psi(\mathbf{u}^{(k)})}. \quad (5.109)$$

If we multiply both numerator and denominator of (5.109) by $\psi(\mathbf{u}^{(k)}) - L_{\pi(1)}$,

$$\frac{1}{\beta_{\pi(1)}^{(k)}} \leq \frac{4 \left\| \mathbf{u}^{(k)} \right\|^2 \left\| \nabla_{\mathbf{u}} \cdot \psi \left(\mathbf{u}^{(k)} \right) \right\|^2}{\psi \left(\mathbf{u}^{(k)} \right) - L_{\pi(1)}} \left(\frac{\psi \left(\mathbf{u}^{(k)} \right) - L_{\pi(1)}}{\psi \left(\mathbf{u}^{(k)} \right)} \right) + \psi \left(\mathbf{u}^{(k)} \right). \quad (5.110)$$

We can identify in this expression the Newton-Raphson step size from (5.10), yielding to the lower bound presented in the Theorem above. ■

Theorem 5.11 (A lower bound for the $\beta_{\pi(2)}$ coefficient)

The non-negative normalized coefficient $\beta_{\pi(2)}^{(k)}$ is lower bounded:

$$\frac{1}{\hat{\beta}_{\pi(2),MIN}^{(k)}} \geq \frac{1}{\beta_{\pi(2)}^{(k)}}, \quad (5.111)$$

with

$$\frac{1}{\hat{\beta}_{\pi(2),MIN}^{(k)}} \triangleq 2 \left\| \mathbf{u}^{(k)} \right\|^2 \left(\mu_{NR}^{(k)} \right)^{-1} + \psi \left(\mathbf{u}^{(k)} \right) + L_{\pi(1)}. \quad (5.112)$$

Where $\mu_{NR}^{(k)}$ is the Newton-Raphson step size from Theorem 5.1.

Proof: The derivation of the lower bound for is shown here. By using the bound (5.106) in (5.107), and noting that $\beta_{\pi(2)}^{(k)} \geq \beta_{\pi(i)}^{(k)}$, for $i \geq 3$, we have

$$\psi \left(\mathbf{u}^{(k)} \right) \leq L_{\pi(1)} + \beta_{\pi(2)}^{(k)} \sum_{i=2}^n L_{\pi(i)}^2 = L_{\pi(1)} + \beta_{\pi(2)}^{(k)} \left(\left\| \mathbf{L}_S \right\|^2 - L_{\pi(1)}^2 \right). \quad (5.113)$$

Therefore, by substituting $\left\| \mathbf{L}_S \right\|^2$ with the help of the relation (5.102), the lower bound for $\beta_{\pi(2)}^{(k)}$, namely $\hat{\beta}_{\pi(2),MIN}^{(k)}$, results:

$$\frac{1}{\beta_{\pi(2)}^{(k)}} \leq \frac{4 \left\| \mathbf{u}^{(k)} \right\|^2 \left\| \nabla_{\mathbf{u}} \cdot \psi \left(\mathbf{u}^{(k)} \right) \right\|^2 + \psi \left(\mathbf{u}^{(k)} \right)^2 - L_{\pi(1)}^2}{\psi \left(\mathbf{u}^{(k)} \right) - L_{\pi(1)}}. \quad (5.114)$$

By using the special binomial product

$$\psi \left(\mathbf{u}^{(k)} \right)^2 - L_{\pi(1)}^2 = \left(\psi \left(\mathbf{u}^{(k)} \right) - L_{\pi(1)} \right) \left(\psi \left(\mathbf{u}^{(k)} \right) + L_{\pi(1)} \right), \quad (5.115)$$

we obtain

$$\frac{1}{\beta_{\pi(1)}^{(k)}} \leq \frac{4 \|\mathbf{u}^{(k)}\|^2 \|\nabla_{\mathbf{u}} \psi(\mathbf{u}^{(k)})\|^2}{\psi(\mathbf{u}^{(k)}) - L_{\pi(1)}} + \psi(\mathbf{u}^{(k)}) + L_{\pi(1)}. \quad (5.116)$$

From this expression, one can identify again the Newton-Raphson step size (5.10) to obtain the lower bound for $\beta_{\pi(2)}^{(k)}$.

These upper and lower bounds for the non-negative normalized coefficients can be used to derive practical step sizes that satisfy the Theorems of Section 5.4.3.

A practical estimator for the μ_P step size

By substituting (5.112) in the value $\beta_{\pi(2)}^{(k)}$ of condition (5.83), a practical estimator for $\mu_P^{(k)}$ is obtained, that always satisfies the Theorem 5.7:

$$\hat{\mu}_P^{(k)} = \frac{1}{2 \left(\mu_{NR}^{(k)} \right)^{-1} + \frac{L_{\pi(1)}}{2 \|\mathbf{u}^{(k)}\|^2}} < \mu_P^{(k)}. \quad (5.117)$$

Note that all the quantities involved are known at each iteration of the algorithm.

A practical estimator for the μ_S step size

By substituting the lower bounds (5.105) and (5.112), respectively, into the values $\beta_{\pi(1)}^{(k)}$ and $\beta_{\pi(2)}^{(k)}$ of condition (5.88), a practical estimator for $\mu_S^{(k)}$ is obtained, that always satisfies the Theorem 5.8:

$$\hat{\mu}_S^{(k)} = \frac{2 \|\mathbf{u}^{(k)}\|^2}{2 \|\mathbf{u}^{(k)}\|^2 \left(\frac{\psi(\mathbf{u}^{(k)}) - L_{\pi(1)}}{\psi(\mathbf{u}^{(k)})} + 1 \right) \left(\mu_{NR}^{(k)} \right)^{-1} + \psi(\mathbf{u}^{(k)}) + L_{\pi(1)}} < \mu_S^{(k)}, \quad (5.118)$$

that can be rewritten as

$$\hat{\mu}_S^{(k)} = \frac{1}{\frac{1}{\mu_{NR}^{(k)}} + \frac{1}{\mu_0^{(k)}}} < \mu_S^{(k)}, \quad (5.119)$$

where $\mu_{NR}^{(k)}$ is the Newton-Raphson step size, and $\mu_0^{(k)}$ is

$$\mu_0^{(k)} = \frac{1}{\frac{\psi(\mathbf{u}^{(k)}) - L_{\pi(1)}(\mu_{NR}^{(k)})^{-1}}{\psi(\mathbf{u}^{(k)})} + \frac{\psi(\mathbf{u}^{(k)}) + L_{\pi(1)}}{2\|\mathbf{u}^{(k)}\|^2}}. \quad (5.120)$$

The step size (5.118) is always below the maximum step size that satisfy condition (5.81), so it guarantees the global monotonous convergence without sacrificing the speed of the algorithm. That makes $\hat{\mu}_S^{(k)}$ our recommended step size for the BCA extraction algorithm.

Note from (5.119) that $\hat{\mu}_S^{(k)} \leq \mu_{NR}^{(k)}$ and $\hat{\mu}_S^{(k)} \leq \mu_0^{(k)}$. When the algorithm starts, the proposed step size $\hat{\mu}_S^{(k)}$ has a value similar to $\mu_0^{(k)}$. However, when the iterations are executed and the algorithm begins to converge, it tends towards the value of the Newton-Raphson step size, which may be unstable at the first iterations.

5.5 Computer simulations

We made some computer simulations, in order to illustrate the results presented in previous sections.

- In the first experiment, a set of $n = 3$ complex QPSK signals (bounded sources) were considered. The length of each source was $T = 1000$. These signals were mixed in a memory-less MIMO system yielding to $m = 3$ observations. The entries of the mixing matrix were randomly generated from a Gaussian distribution. The observations were pre-whitened before running 50 iterations of the BCA extraction algorithm described in Chapter 4.

In Figure 5.6, the value of the relative remaining cost $(\psi(\mathbf{u}^{(k)}) - \psi_{\min}) / \psi_{\min}$ during the execution of the algorithm is shown. All the step sizes used in these simulations were computed by its estimators, presented in the previous section.

By using the Newton-Raphson step size, we obtain the red curve. As can be seen in the figure, the convergence is very fast but this value does not guarantee a global monotonous convergence, so it is not a recommended option. Indeed, we have selected an example where this step size converges to a solution far from the basin where it was

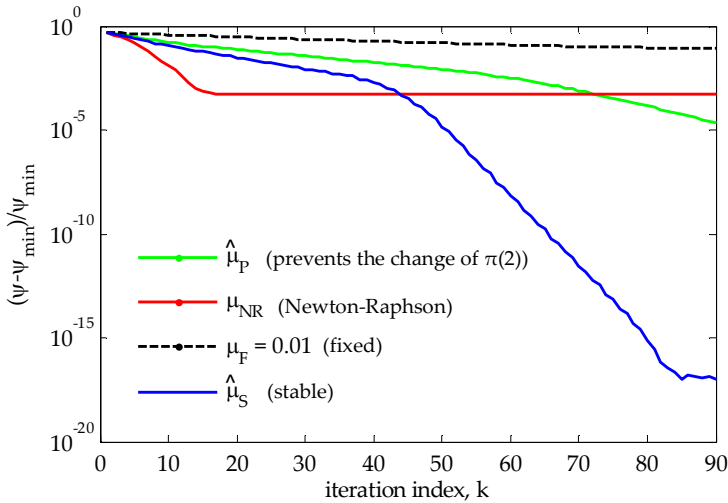


Figure 5.6 – First experiment. We show the remaining value of the normalized cost function, for four different step sizes. The NR step size does not converge in this example. The blue line corresponds to $\hat{\mu}_S^{(k)}$, the red line to the Newton-Raphson step size, and the green line to $\hat{\mu}_P^{(k)}$. The black dashed line corresponds to a fixed step size of value 0.01.

initialized. The step size proposed in (5.118) yields to the blue curve. The corresponding algorithm is not as fast as the previous one, but it does guarantee the global monotonous convergence, so it is the recommended option.

The execution of the algorithm using the step size (5.117) yields to the green curve. Although this value also guarantees the global convergence, the iteration is slow when compared to the maximum step size. Finally, a fixed step size of value $\mu_F = 0.01$ is presented for comparison. It can be seen that the corresponding algorithm is very slow and also does not guarantee the convergence to a local solution.

- The second experiment is conducted with $n = m = 2$ for the ease of the graphical representation. The Figure 5.7 shows the movement of the g_1 and g_2 coefficients on the two dimensional plane, depending on where the algorithm is initialized. As in the example illustrated in Figure 5.4, the perimeter of the source 2 is greater than the perimeter of the source 1, and there is a change of the direction of convergence that corresponds to a saddle point of the cost function.

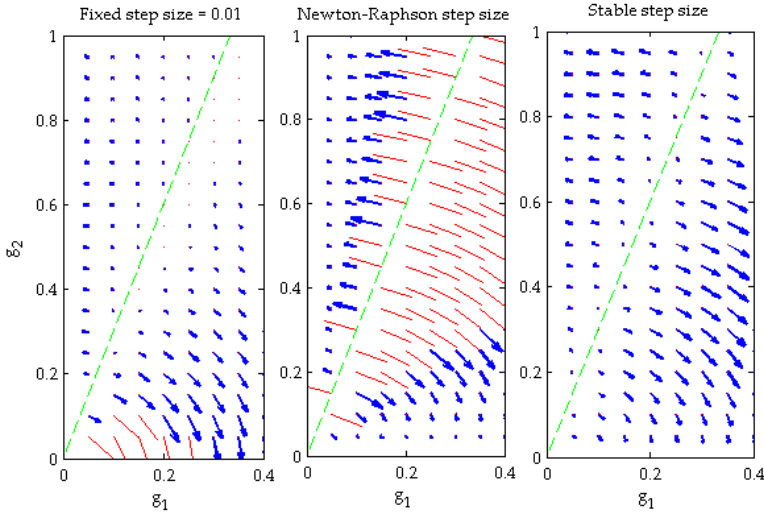


Figure 5.7 – Second experiment. The blue arrows show the behavior of the g_1 and g_2 coefficients on the two dimensional plane, during several executions of the first iterations of the Bounded Component Analysis extraction algorithm. (left) A fixed step size $\mu_F = 0.01$ is used. (center) The Newton-Raphson step size is used. (right) The step size $\mu_S^{(k)}$ is used. The straight red lines correspond to the points of initialization where the movement is so strong to guarantee the convergence. The green dashed line corresponds to the direction of the vector \mathbf{L}_S , and divides the plane in two regions of convergence.

Three types of step sizes were compared. In the left figure, a fixed value of $\mu_F = 0.01$ is used, which results in non-stable points near the origin (the step size cannot adapt to the thin place where it is descending). The center figure corresponds to the execution of the algorithm with the Newton-Raphson step size. Points far from a local minimum have a large value for $\mu_{NR}^{(k)}$, so the monotonous convergence is not guaranteed. Finally, the right figure shows the plane when the proposed step size (5.118) is used. All of the points of the plane converges (in the sense given by (5.81)) to an axis, thus extracting the corresponding source. This corroborates the behavior observed in the Figure 5.4.

- The third experiment consists on the extraction of a bounded communication signal (BPSK, QPSK or 32-QAM) from a mixture of $n = m = 10$ observations and sources. The Figure 5.8 studies the values of the three $\beta_i^{(k)}$ coefficients (for the ease of the interpretation, they have not been ordered).

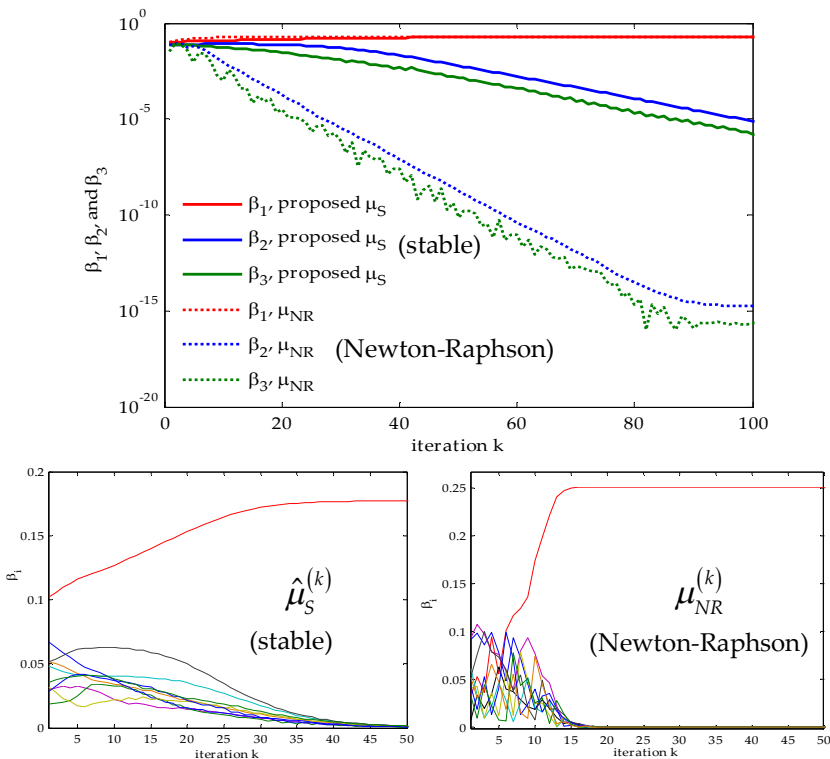


Figure 5.8 – Third experiment. This figure shows the behavior of the $\beta_i^{(k)}$ coefficients during the 100 iterations of the BCA extraction algorithm. (upper) Comparison between the $\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}$ coefficients when using the proposed step size (solid lines), and when using the Newton-Raphson step size (dotted lines). (lower left) The movement of the ten $\beta_i^{(k)}$ coefficients when using the proposed step size. (lower right) The movement of the ten $\beta_i^{(k)}$ coefficients when using the Newton-Raphson step size.

In the upper figure, it can be seen that the Newton-Raphson step size is faster than the proposed solution. However, as it is shown in the lower figures, the proposed step size prevents the change of the $\pi(1)$ coefficient (the red line that grows and stabilizes at a non-zero value), thus ensuring that the extracted source is the one that is located at the bottom of the basin where the algorithm was initialized.

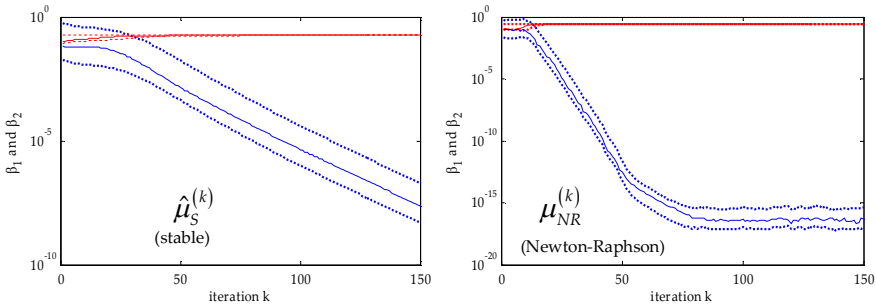


Figure 5.9 – Third experiment. This figure shows the convergence of the Newton-Raphson step size (right) and the proposed step size (left).

In addition, in the Figure 5.9, the upper and lower bounds for the step sizes are shown. The lower bounds $\hat{\beta}_{i,MIN}^{(k)}$ can be used as estimators for the $\beta_i^{(k)}$ parameters, and the resulting practical step size $\hat{\mu}_S^{(k)} < \mu_S^{(k)}$ guarantees the monotonous convergence of the algorithm.

5.6 Conclusions

This Chapter addresses the study of the global monotonous convergence of the BCA extraction algorithm. We started by studying the non-differentiable points and the special shape of the optimization surface. From here, we focus on the analysis of the update equation and on the selection of appropriate step sizes. The main results of this study are summarized below:

- We derive first and second order local approximations to the shape of the cost function, which yields to the fast (although potentially unstable) options known as the Newton-Raphson step size and the step size that uses the second order information.
- After the analysis of the geometry of the update equation, we set three conditions in the pursuit of specific behaviors of the iterations. These conditions lead to three upper bounds for the value of the step size, but they depend on unknown values so they cannot be used in practice. One of these bounds guarantees the global monotonous convergence of the algorithm, in the sense of the descent of a certain coefficient.
- Practical step sizes must be proposed if one aims to use the results of the convergence analysis in practical algorithms. By using realistic bounds, we derive these practical step sizes, whose structure reveals

dependence with the well-known Newton-Raphson step size.

Three computer experiments shows how the parameters presented during the study of convergence behaves in practical applications. The proposed step size is satisfactorily fast and also guarantees the monotonous convergence to a local minimum, wherever it is initialized.

6 BOUNDED COMPONENT ANALYSIS OF THE TRAINING ERROR

*I never teach my pupils; I only attempt to provide the
conditions in which they can learn.*

- A. Einstein -

In this chapter, we aim to derive an alternate framework for the Bounded Component Analysis extraction algorithm. In addition to the blind model, there is also a supervised model for the problem of source extraction, which introduces subtle differences and similitudes with the unsupervised criteria presented in previous chapters.

6.1 Introduction

This chapter reviews our results for the problem of supervised signal extraction from linear mixtures of bounded sources, mainly based on the works (Cruces, 2011) and (Aguilera-Bonet, 2012). We assume a short training sequence (only for the source that we want to extract) and an under-determined noisy mixture.

We introduce a cost function based on the convex perimeter of the training error between the estimate of the desired source and the training sequence. We also show that one can achieve a good estimate of the desired signal by minimizing the previous cost function.

The problem can be motivated by the interference cancellation in wireless communications systems. The desired signal is the information that comes from the antenna of a single user, and the remaining sources are treated as interference from other users. Those signals consist on a sequence of complex-valued symbols that are drawn from a random process whose probability density function is represented by a constellation. Due to the multiple accesses to the medium, these systems tend to operate in interference limited scenarios and the mixture usually becomes *under-determined*.

The receiver consists on an array of m antennas, and in real scenarios the number of them can be lower than the number of transmitters n , turning the problem into under-determined. There are some existing techniques that can work for this kind of problems like the MMSE, *Zero Forcing (ZF)* or *sphere decoding* (see (Damen, 2003) and (Shahbazpanahi, 2006)), but they require the exact knowledge of the mixture matrix or of the training sequence for all the sources, information which we assume unavailable.

The chapter is organized as follows. In Section 6.2, the mixture model and the concept of training error are presented. Section 6.3 introduces under-determined mixtures, which yields to imperfect extractions due to the lack of degrees of freedom. In Section 6.4, the BCA framework is extended to supervised problems, which yields to the algorithm studied in Section 6.5.

The main differences and similitudes of the proposed criterion with the MMSE criterion are highlighted in Section 6.6. Some of these characteristics are based upon the geometry of the norms involved in each option, which are studied in Section 6.7. In Section 6.8, we add a constraint on the optimization problem to solve the issues related to the presence of noise. Finally, Section 6.9 presents some computer simulations to show the performance of the proposed method, and Section 6.10 condenses the conclusions to this chapter.

6.2 Signal Model

In this section, we establish the model used for the *supervised* or *semi-blind* version of the Bounded Component Analysis extraction algorithm. This means that we use a small training sequence to solve the extraction problem. When this is done and if the model does not change, we can use the same solution to extract unknown signals.

The semi-blind criterion proposed here shares many similarities with the blind version of BCA presented in the previous chapter, but their differences force us to introduce the new model.

As in the blind case, consider a sequence of observations, drawn from a stationary random vector process,

$$\mathbf{x}(k) = \begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix} \in \mathbb{C}^m. \quad (6.1)$$

We assume that this vector of observations admits the following linear model,

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{q}(k), \quad (6.2)$$

where

$$\mathbf{s}(k) = \begin{bmatrix} s_1(k) \\ \vdots \\ s_n(k) \end{bmatrix} \in \mathbb{C}^n \quad (6.3)$$

represents a vector of n bounded sources, $\mathbf{A} \in \mathbb{C}^{m \times n}$ is the mixing matrix and

$$\mathbf{q}(k) = \begin{bmatrix} q_1(k) \\ \vdots \\ q_m(k) \end{bmatrix} \in \mathbb{C}^m \quad (6.4)$$

is a bounded¹⁶ noise vector.

The linear signal extraction problem consists on determining the extraction

¹⁶ For the sake of simplicity, we assumed in the theoretical derivations of the proposed approach that the noise is bounded. However, in practice one can relax in part this assumption because for most of the practical distributions, the empirical support of a finite set of samples is almost always bounded.

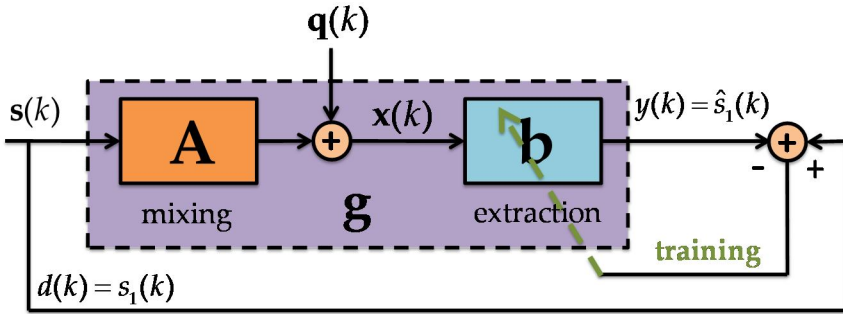


Figure 6.1 – This block diagram shows the signals and the systems presented in the training stage of a supervised signal extraction algorithm.

vector that leads to the best linear estimate, in some given sense, of the signal of interest,

$$y(k) = \mathbf{b}^H \mathbf{x}(k) \in \mathbb{C}. \quad (6.5)$$

As in the blind scenario, $\mathbf{b} \in \mathbb{C}^m$ is the extraction vector. Introducing the residual mixture vector, $\mathbf{g} = \mathbf{A}^H \mathbf{b} \in \mathbb{C}^n$, we can express the output as a linear combination of sources and noise,

$$y(k) = \mathbf{g}^H \mathbf{s}(k) + \mathbf{b}^H \mathbf{q}(k). \quad (6.6)$$

Hereinafter we assume, without loss of generality, that this signal of interest is the first one, so $y(k) \triangleq \hat{s}_1(k)$. This estimate is computed from the observations, the model and all our prior knowledge of the signal of interest, which in our case includes a short training sequence, $s_1(0), \dots, s_1(T)$. When the length T is small compared to the entire length of the sequences, the method is called *semi-supervised* or semi-blind.

The main difference between this supervised method and the blind method comes here. The next concept is essential to derive the criterion needed in the supervised version of the BCA extraction method.

Definition (Training error): The estimation error (or *training error*) is defined as the difference between the desired signal and the estimated signal,

$$e(k) = d(k) - y(k) = s_1(k) - \hat{s}_1(k) = s_1(k) - \mathbf{g}^H \mathbf{s}(k) - \mathbf{b}^H \mathbf{q}(k). \quad (6.7)$$

In Figure 6.1, an overview of the model used in this chapter is shown. The extraction vector \mathbf{b} is synthesized by using the training error. This can be

analyzed by components,

$$e(k) = (1 - g_1^*)s_1(k) - g_2^*s_2(k) - \dots - g_n^*s_n(k) - b_1^*s_1(k) - \dots - b_m^*q_m(k). \quad (6.8)$$

Or by grouping in vector form,

$$y(k) = (\boldsymbol{\delta}_1 - \mathbf{g})^H \mathbf{s}(k) - \mathbf{b}^H \mathbf{q}(k) = \sum_{i=1}^n (\boldsymbol{\delta}_1 - \mathbf{g}^*)_i s_i(k) - \sum_{j=1}^m b_j^* q_j(k), \quad (6.9)$$

where $\boldsymbol{\delta}_1 = [1, 0, \dots, 0]^T$ is the unit vector. From (6.9), one can observe two contributions to the error, one coming from the residual interference due to the departure of the residual mixture vector \mathbf{g} from the ideal vector $\boldsymbol{\delta}_1$, and another coming from the additive noise component in the linear estimate.

6.3 Under-determined Mixtures

As we use a small training sequence, we have some advantages over the blind criterion. However, we can complicate the model to exploit the extra information provided by the *pilot symbols* in that training sequence.

The first addition to the model was the inclusion of noise, represented by the vector of random processes $\mathbf{q}(k)$. The effect of the noise can amplify or cancel the value of the extraction vector, so it must be treated carefully.

In addition to the presence of noise, this time we work with an under-determined mixture. We assume therefore more sources than sensors ($n > m$), so the columns of the matrix \mathbf{A} constitute an over-complete basis of \mathbb{C}^m , and thus the mixture is non-invertible. Therefore, the perfect recovery of $s_1(k)$ is not possible, in general, and one can only expect a recovery $y(k)$ that contains a certain amount of additive noise and interference from the other sources.

Among the existing approaches aimed to solve under-determined problems, there are blind techniques like (Comon, 2004) or (Zarzoso, 2006) that restore the separability of the sources by exploiting their discrete nature and the knowledge of their alphabet. The solution of the problem can be regularized if one assumes, additionally, the prior knowledge of a training sequence like we do in this chapter. This assumption is currently plausible in most of the digital communications standards.

However, supervised identification techniques like standard least squares (Shahbazpanahi, 2006 p. 3583) or Zero Forcing (Johnson Jr, 1995), still require training sequences for all the sources and they are not usually suitable in under-determined scenarios. Also, sphere decoding techniques can be used to solve

under-determined problems, but they require the knowledge of the channel state information (Damen, 2003 p. 2397). On the contrary, our proposal aims to exploit the bounded latent structure of the observations together with some partial knowledge of the desired source, only in the form of one training sequence.

Note that any under-determined mixture with m observations and $n > m$ sources can be transformed into a determined mixture, by simply assuming that the contribution at the observations of the $n - m$ uninteresting sources forms part of the additive noise. We do not prefer this latter notation because it hides part of the statistical known structure of the noise.

The following sections extend BCA for noisy under-determined mixtures and semi-supervised problems, with the aim to reveal the bounded component structure of the training error.

6.4 Bounded Component Analysis of the Training Error

Let us unify the notation of the bounded components of the estimation error. By examining (6.9), the following vectors can be defined.

Definition (Extended vector of coefficients): The *extended vector of coefficients* groups the coefficients corresponding to the residual mixture, and those corresponding to the noise term,

$$\mathbf{g}' = \begin{bmatrix} \delta_1 - \mathbf{g} \\ -\mathbf{b} \end{bmatrix} \in \mathbb{C}^{n+m}. \quad (6.10)$$

Definition (Extended vector of sources): The *extended vector of sources* groups the sequences of sources and noise, both of bounded convex support,

$$\mathbf{s}'(k) = \begin{bmatrix} \mathbf{s}(k) \\ \mathbf{w}(k) \end{bmatrix} \in \mathbb{C}^{n+m}. \quad (6.11)$$

Thus, (6.9) can be rewritten as

$$e(k) = (\mathbf{g}')^H \mathbf{s}'(k) = \sum_{i=1}^{n+m} (g'_i)^* s'_i(k). \quad (6.12)$$

As a BCA method, the supervised source extraction algorithm is based on the concepts of geometry presented in Chapter 4. Indeed, we are going to quantify the estimation error using its convex support.

By minimizing the convex support S_e , we are minimizing the error $e(t)$ itself and, therefore, making the estimate $y(t)$ closer to the desired signal $\hat{s}_1(t)$. This is similar to the idea behind the MMSE method, in which the estimation becomes more accurate by minimizing the power or the variance of the training error.

This time, the main assumptions of BCA operates on the vector of extended sources. Thus, the Theorem 4.2 states that the Cartesian decomposition of the joint convex support of this extended vector of sources, i.e.,

$$S_{s'} = S_{s'_1} \times \cdots \times S_{s'_{n+m}}, \quad (6.13)$$

implies the direct decomposition of the convex hull of the error:

$$S_e = S_{(g'_1)^* s'_1} \oplus \cdots \oplus S_{(g'_{n+m})^* s'_{n+m}}, \quad (6.14)$$

where the operator \oplus denotes the Minkowski sum between sets.

6.4.1 The convex perimeter functional as a cost function

This section introduces a convenient measure of error that will be used as a cost function in the next sections. The convex perimeter of the training error is defined in the way shown in Section 4.5. It is the perimeter of the convex hull of the support set of the random variable, i.e., $L(e(k)) \equiv \mathcal{L}(S_{e(k)})$.

Also, it has the same properties of additivity, homogeneity and invariance under translations and rotations presented in Theorem 4.4. By applying these properties to (6.14), we obtain the expression that relates the perimeter of the error with the perimeter of the extended sources,

$$\psi(\mathbf{b}) = L(e(k)) = \sum_{i=1}^{m+n} |g'_i| L(s'_i(k)) \in \mathbb{C}. \quad (6.15)$$

So, the optimization problem is

$$\mathbf{b}_{opt} = \arg \min_{\mathbf{b}} \{L(e(k))\}. \quad (6.16)$$

After rewriting the result in terms of the sources and the noise, we obtain

$$L(e(k)) = \sum_{i=1}^n |(\delta_1 - \mathbf{g})_i| L(s_i(k)) + \sum_{j=1}^m |b_j| L(q_j(k)). \quad (6.17)$$

This reveals that the convex perimeter of the error is a weighted 1-norm of the mismatch vector $(\delta_1 - \mathbf{g})$, plus a nuisance term coming from the convex

perimeter of the components of the noise.

6.4.2 Structure of the convex perimeter

Suppose that \mathbf{D} is a diagonal matrix whose elements are $L(s'_i(k))$, for $i = 1, \dots, m+n$,

$$\mathbf{D} = \begin{bmatrix} L(s'_1(k)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & L(s'_{m+n}(k)) \end{bmatrix}. \quad (6.18)$$

We define the *auxiliary affine function* of the extraction vector,

$$G(\mathbf{b}) = \mathbf{D} \left(\delta_1 - \begin{bmatrix} \mathbf{A}^H \\ \mathbf{I}_m \end{bmatrix} \mathbf{b} \right) = \mathbf{D} \begin{bmatrix} \mathbf{1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} a_{11}^* & \cdots & a_{m1}^* \\ \vdots & \ddots & \vdots \\ a_{1n}^* & \cdots & a_{mn}^* \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \mathbf{b} \in \mathbb{C}^{m+n}. \quad (6.19)$$

Using this, we can rewrite the convex perimeter (6.17) as the composition of the 1-norm function and the auxiliary affine function, i.e.,

$$L(e(k)) = \|G(\mathbf{b})\|_1. \quad (6.20)$$

The 1-norm and $G(\cdot)$ are both continuous and piece-wise affine functions of their arguments, thus, these properties are inherited by $L(e(k))$. Moreover, this means that the convex perimeter $L(e(k))$ is a convex function of the extraction vector (see (Boyd, 2009)).

The 1-norm minimization subject to linear/affine constraints is one of the preferred approaches in the field of *Compressed Sensing* for the pursuit of sparse solutions of under-determined systems of linear equations. See (Donoho, 2006) for an introduction on this promising area.

The properties of the 1-norm are responsible for most of the features of the solutions obtained by the minimum convex perimeter criterion. At this point, we are able to present a convenient algorithm that minimizes the cost function.

6.5 The Supervised BCA Extraction Algorithm

In this section, we present the gradient descent algorithm used to minimize the criterion of minimum convex perimeter (6.16).

6.5.1 The gradient of the convex perimeter

The criterion for the supervised extraction of a bounded source is the minimization of the convex perimeter of the training error. The proposed descent algorithm needs the derivation of the gradient of the cost function. One can apply the rules of Wirtinger Calculus to

$$\psi(\mathbf{b}) = \sum_{i=1}^V |\Delta e_i|, \quad (6.21)$$

where Δe_i are the edges between samples of the convex hull of the error,

$$\Delta e_i = e(\zeta_i) - e(\zeta_{i-1}), \quad (6.22)$$

with $[\zeta_0, \dots, \zeta_V] = \text{conv}\{e\}$.

Theorem 6.1 (Gradient of the convex perimeter)

Given the cost function (6.21), the gradient needed for the update of the gradient descent algorithm is computed as

$$\nabla_{\mathbf{b}} \psi(\mathbf{b}) = - \sum_{i=1}^V \frac{\Delta e_i^* \Delta \mathbf{x}_i}{2|\Delta e_i|}. \quad (6.23)$$

The segments Δe_i and $\Delta \mathbf{x}_i$ are computed from (6.22) and $\Delta \mathbf{x}_i = \mathbf{x}(\zeta_i) - \mathbf{x}(\zeta_{i-1})$, respectively. Note how the only convex hull needed is $\text{conv}\{e\}$.

Proof: We start with the expression

$$e(k) = d(k) - y(k) = d(k) - \mathbf{b}^H \mathbf{x}(k). \quad (6.24)$$

The gradient of the cost function is

$$\nabla_{\mathbf{b}} \psi(\mathbf{b}) = \nabla_{\mathbf{b}} L(e) = \sum_{i=1}^V \nabla_{\mathbf{b}} |\Delta e_i|. \quad (6.25)$$

The gradients of the edges Δe_i and Δe_i^* are

$$\nabla_{\mathbf{b}^*} (\Delta e_i) = -\nabla_{\mathbf{b}^*} (\mathbf{b}^H \Delta \mathbf{x}_i) = -\Delta \mathbf{x}_i, \quad (6.26)$$

$$\nabla_{\mathbf{b}^*} (\Delta e_i^*) = -\nabla_{\mathbf{b}^*} (\mathbf{b}^T \Delta \mathbf{x}_i^*) = \mathbf{0}. \quad (6.27)$$

By applying Wirtinger Calculus on the moduli $|\Delta e_i|$,

$$\begin{aligned} \nabla_{\mathbf{b}^*} |\Delta e_i| &= \nabla_{\mathbf{b}^*} (\Delta e_i \Delta e_i^*)^{1/2} \\ &= \nabla_{\Delta e_i^*} (|\Delta e_i|) \underbrace{\nabla_{\mathbf{b}^*} (\Delta e_i^*)}_0 + \nabla_{\Delta e_i} (|\Delta e_i|) \underbrace{\nabla_{\mathbf{b}^*} (\Delta e_i)}_{-\Delta \mathbf{x}_i}. \end{aligned} \quad (6.28)$$

The first term is zero because of (6.27). By substituting the gradient

$$\nabla_{\Delta e_i^*} (|\Delta e_i|) = \frac{\Delta e_i^*}{2|\Delta e_i|} \quad (6.29)$$

into (6.28), we obtain

$$\nabla_{\mathbf{u}^*} (|\Delta e_i|) = -\frac{\Delta e_i^* \Delta \mathbf{x}_i}{2|\Delta e_i|}. \quad (6.30)$$

Finally, the summation of all the gradients of the segments results in (6.23). ■

In Table 6.1, the pseudo-code of supervised BCA extraction algorithm is shown.

6.5.2 The step size

As in the previous chapter, it is important to select a proper step size. The derivation of the expressions for the step size follows the same method that for the blind criterion. That is, the first and second order approximations of the local behavior of the cost function.

The Newton-Raphson step size

The Newton-Raphson (NR) step size is very similar to the blind version,

$$\mu = \frac{\psi(\mathbf{b})}{2 \|\nabla_{\mathbf{b}^*} \psi(\mathbf{b})\|^2}, \quad (6.31)$$

<i>Inputs</i>	$\mathbf{x}(k)$: whitened observations $d(k)$: training reference		
<i>Repeat until convergence</i>	$e(k) = d(k) - \mathbf{b}^H \mathbf{x}(k)$: training error		
	<i>Computation of the gradient</i>	$[\zeta_0, \dots, \zeta_V] = \text{conv}\{e\}$	
		<i>Repeat for</i> $i = 1, \dots, V$	$\Delta \mathbf{x}_i = \mathbf{z}(\zeta_i) - \mathbf{z}(\zeta_{i-1})$ $\Delta e_i = y(\zeta_i) - y(\zeta_{i-1})$
		$\nabla_{\mathbf{b}, \psi}(\mathbf{b}) = -\sum_{i=1}^V \frac{\Delta e_i^* \Delta \mathbf{x}_i}{2 \Delta e_i }$	
μ : adaptive step size $\mathbf{b} \leftarrow \mathbf{b} - \mu \nabla_{\mathbf{b}, \psi}(\mathbf{b})$: update $\mathbf{b} \leftarrow \mathbf{b} / \ \mathbf{b}\ $: normalization increase the iteration index			
<i>Outputs</i>	\mathbf{b} : extraction vector $y(k) = \mathbf{b}^H \mathbf{x}(k)$: estimate of the source		

Table 6.1: Pseudo-code of a supervised BCA extraction algorithm.

where the gradient is computed from (6.23). The only difference comes from the fact that $\psi_{\min} = 0$ for the supervised version, because we aim to cancel out the error sequence in a perfect separation.

A step size that uses the second order information

This step size for the supervised gradient descent algorithm is given by the same expression that for the blind version (5.16), but taking again $\psi_{\min} = 0$.

$$\mu = \frac{\|\nabla_{\mathbf{b}, \psi}\|^2 - \sqrt{\|\nabla_{\mathbf{b}, \psi}\|^4 - 2(\nabla_{\mathbf{b}, \psi})^H \mathbf{H}_{\mathbf{b}\mathbf{b}} \psi (\nabla_{\mathbf{b}, \psi}) \psi}}{(\nabla_{\mathbf{b}, \psi})^H \mathbf{H}_{\mathbf{b}\mathbf{b}} \psi (\nabla_{\mathbf{b}, \psi})}, \quad (6.32)$$

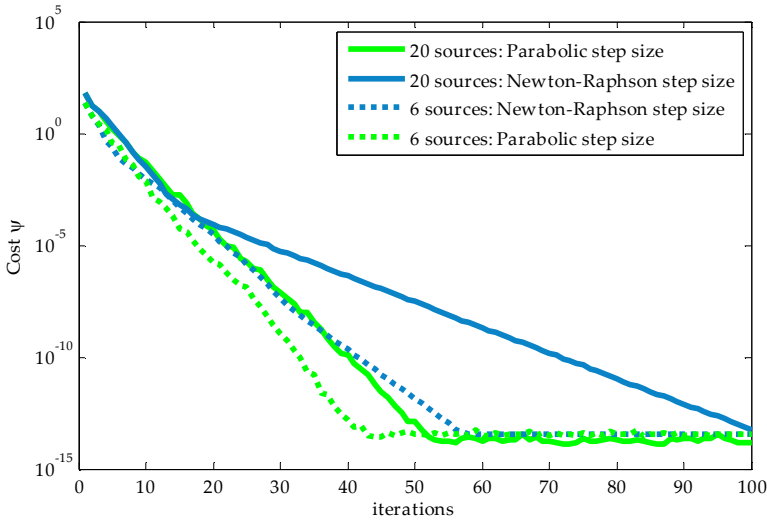


Figure 6.2 – This figure shows, in logarithmic scale, the comparison of the value of the cost function $\psi(\mathbf{b})$, when the Newton-Raphson (blue) and the step size that uses the second-order information (green) are used in a supervised extraction. We study two different cases: having 6 QPSK sources (dotted) and having 20 QPSK sources (solid).

where

$$\underline{\mathbf{H}}_{\mathbf{bb}}\psi(\mathbf{b}) = \begin{bmatrix} \mathbf{H}_{\mathbf{b}} & \tilde{\mathbf{H}}_{\mathbf{b}} \\ \tilde{\mathbf{H}}_{\mathbf{b}}^* & \mathbf{H}_{\mathbf{b}}^* \end{bmatrix} \in \mathbb{C}^{2m \times 2m}, \quad (6.33)$$

is the augmented Hessian. To obtain practical expressions, we use the following complex derivatives:

$$\frac{\partial |\Delta e_i|}{\partial \mathbf{b}^*} = \frac{\partial (\Delta e_i \Delta e_i^*)^{1/2}}{\partial \mathbf{b}^*} = -\frac{\Delta e_i^* \Delta \mathbf{x}_i}{2|\Delta e_i|} \in \mathbb{C}, \quad (6.34)$$

$$\frac{\partial |\Delta e_i|}{\partial \mathbf{b}} = \frac{\partial (\Delta e_i \Delta e_i^*)^{1/2}}{\partial \mathbf{b}} = -\frac{\Delta e_i \Delta \mathbf{x}_i^*}{2|\Delta e_i|} \in \mathbb{C}. \quad (6.35)$$

The sub-matrices of (6.33) are computed by applying the operators $\partial(\cdot)/\partial \mathbf{b}$ and $\partial(\cdot)/\partial \mathbf{b}^*$ to the transpose of the gradient of the cost function (6.23):

$$\mathbf{H}_b = \frac{\partial}{\partial \mathbf{b}} \left(\frac{\partial L(e)}{\partial \mathbf{b}^*} \right)^T = \sum_{i=1}^V \frac{\partial}{\partial \mathbf{b}} \left(-\frac{\Delta e_i^* \Delta \mathbf{x}_i^T}{2|\Delta e_i|} \right) = \sum_{i=1}^V \frac{\Delta \mathbf{x}_i^* \Delta \mathbf{x}_i^T}{4|\Delta e_i|} \in \mathbb{C}^{m \times m}, \quad (6.36)$$

$$\tilde{\mathbf{H}}_b = \frac{\partial}{\partial \mathbf{b}^*} \left(\frac{\partial L(e)}{\partial \mathbf{b}^*} \right)^T = \sum_{i=1}^V \frac{\partial}{\partial \mathbf{b}^*} \left(-\frac{\Delta e_i^* \Delta \mathbf{x}_i^T}{2|\Delta e_i|} \right) = -\sum_{i=1}^V \frac{\Delta \mathbf{x}_i \Delta \mathbf{x}_i^T (\Delta e_i^*)^2}{4|\Delta e_i|^3} \in \mathbb{C}^{m \times m}. \quad (6.37)$$

In the Figure 6.2, the value of the cost function $\psi(\mathbf{u})$ in a computer simulation is shown for the two step sizes proposed in this section.

The parameters are the same used in the simulations of Section 5.3 in the previous chapter. We used $n=m=6$ QPSK sources of length $T=1000$, and $n=m=20$ QPSK sources of length $T=5000$. It can be seen that the gain between both step sizes is greater for high dimensionality problems.

As in the blind case, the step size that uses the second order information achieves a faster convergence than the Newton-Raphson step size (in the sense of the number of iterations), although the computational complexity is greater in the former case.

6.6 Structural Similitudes and Differences with the Mean Square Error

The convex perimeter of the training error is a good measure of the accuracy of the estimation. In this section, we compare the structure of (6.17) with a well-known approach used in signal processing.

For sources and noises which are spatially uncorrelated and mutually uncorrelated, it is well known that the Mean Square Error of the linear estimate is given by

$$E\{|e(k)|^2\} = \sum_{i=1}^n |(\boldsymbol{\delta}_1 - \mathbf{g})_i|^2 \sigma_{s_i}^2 + \sum_{j=1}^m |b_j|^2 \sigma_{q_j}^2, \quad (6.38)$$

where $\sigma_{s_i}^2 = E\{s_i(k)\}$ and $\sigma_{q_i}^2 = E\{q_i(k)\}$ are the variances of the sources and the noise sequences, respectively. This is a weighted 2-norm of the mismatch vector $(\boldsymbol{\delta}_1 - \mathbf{g})$, with the variances acting as weighting coefficients, and a second nuisance term due to the noise.

Let us compare the structure and properties of both functions, the minimum convex perimeter $L(e(k))$ and the variance $E\{|e(k)|^2\}$. First, the main similitudes are:

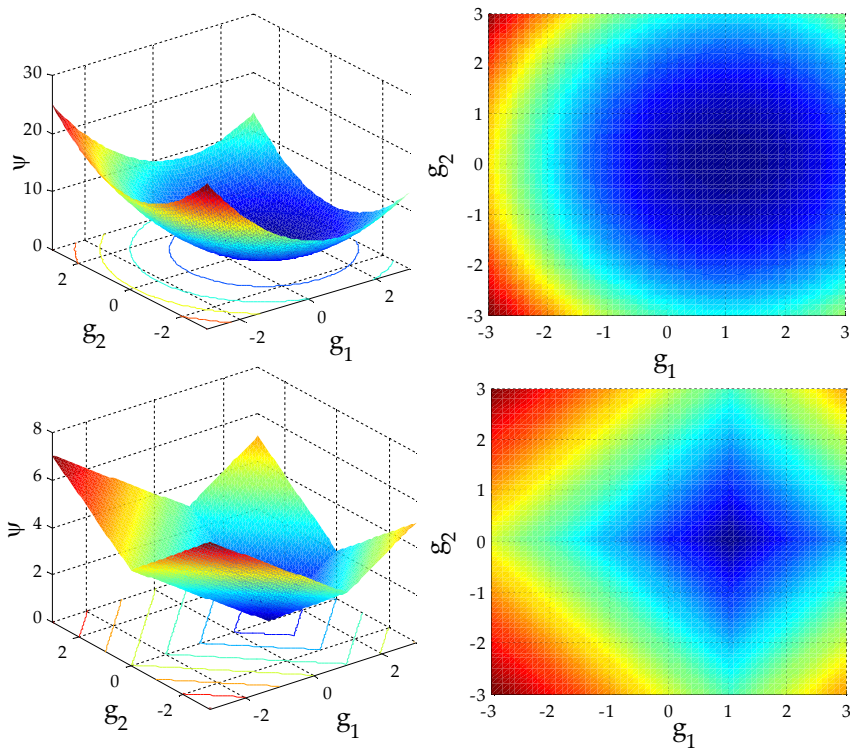


Figure 6.3 - This figure shows the shape of the MSE (upper row) and minimum convex perimeter (lower row) cost functions. On the left column, the 3D representation of the surface ψ is shown. On the right column, a central view of the same function is shown.

- Both functions are composed of weighted norms, and has a term corresponding to the residual mixture of the sources and another term corresponding to the noise.
- Both functions are continuous, convex, and free of local minima.

In spite of these similarities, there are also some key differences:

- There is a closed-form solution for the extraction vector which minimizes the MSE criterion (6.38), which is given by the Wiener solution, presented in Section 3.2. However, there is no closed-form solution for the minimum of the convex perimeter (6.17).
- The convex perimeter function is composed of 1-norms, and the

weights are the perimeters of the sources and the noise. On the other hand, the MSE function is composed of 2-norms, and the weights are the variances of the sources and the noise. The type of the norms involved is key to analyze the behavior of the proposed criterion.

- The criterion of minimum convex perimeter enforces the exact cancellation of some of the interferences, while the MMSE criterion tries to reduce the overall contribution of all of them.
- The function (6.17) pursues a sparse structure on the coefficients of the extended vector of coefficients \mathbf{g}' , while the function (6.38) minimizes the value of all of them but without cancelling completely no one of them.

In the Figure 6.3, examples of the shape of both cost functions are shown, versus the value of the coefficients of the global transfer vector g_1 and g_2 . Note that the minimum of the cost function is reached at the point $(1,0)$. These properties will be highlighted in the rest of the chapter, and most of them will be further explained in the next sections.

6.7 Geometry of the Norms

The 1-norm and 2-norm are important operators in the definition of practical cost functions. In this section we explain the geometry of these norms, and why the cost functions based on them have such interesting behaviors (Elad, 2004).

6.7.1 The 1-norm

As it is well known, the 1-norm of a vector \mathbf{g} is defined as

$$\|\mathbf{g}\|_1 = \sum_{i=1}^n |g_i| \in \mathbb{R}. \quad (6.39)$$

The absolute value produces points where the function is non-differentiable.

The goal is to minimize the cost function (6.15), subject to the linear model (6.12). Without loss of generality, the following analysis is done with $m+n=2$ and $L(s'_1(k)) = L(s'_2(k)) = L_i$ to simplify the explanations.

Let us analyze the structure of the problem,

$$\min_{g'_1, g'_2} \{L(e(k))\} = \min_{g'_1, g'_2} \{|g'_1|L_i + |g'_2|L_i\}, \quad (6.40)$$

$$\text{s.t. } e(k) = (g'_1)^* s'_1(k) + (g'_2)^* s'_2(k). \quad (6.41)$$

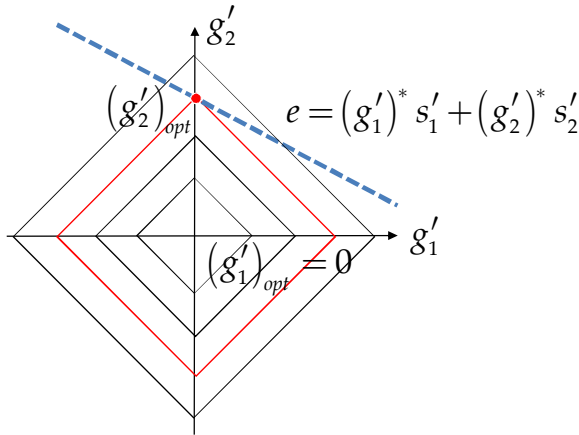


Figure 6.4 - This figure shows the geometry of the 1-norm discussed in this section. The surfaces of equally norm are rhombus. The axes are the coefficients g'_1 and g'_2 .

At each iteration, the values of $L(e(k))$, $L(s'_1(k))$, $L(s'_2(k))$, $s'_1(k)$, and $s'_2(k)$ are fixed. The variables to minimize the cost function are g'_1 and g'_2 .

The constraint (6.41) is an affine function on the plane (g'_1, g'_2) . Also, the function

$$\frac{L(e(k))}{L_i} = |g'_1| + |g'_2|, \quad (6.42)$$

defines a family of level curves on the plane (g'_1, g'_2) .

In Figure 6.4, a representation of this plane is shown. The dashed line represents the constraint (6.41). The solution to this problem, namely $(g'_1, g'_2)_{opt}$, must be a point over this line. The family of rhombus on the plane is the geometry locus of the points that satisfy (6.42). The radius of these rhombus are defined by the value of $L(e(k))/L_i$.

The solution to the optimization problem is marked by a red dot, and is given by the point of the dashed line that belongs to the rhombus of minimum normalized perimeter. No other point of the constraint line is nearer to the origin, in 1-norm sense.

The figure explains graphically why the 1-norm pursues sparse solutions for the extended vector of coefficients. A point that belongs to the constraint line and simultaneously to the rhombus of minimum normalized perimeter must be on

one of the axes.

This means that one of the coefficients of the vector \mathbf{g}' is zero and the other is non-zero, thus providing a sparse solution. In general, all norms $\|\mathbf{g}\|_p$, with $0 \leq p \leq 1$, pursue sparse solutions because the point of minimum norm is located on an axis.

6.7.2 The 2-norm

It is time to analyze the 2-norm in the same way as we did for the 1-norm. The 2-norm of a vector \mathbf{g} is defined as

$$\|\mathbf{g}\|_2 = \sum_{i=1}^n |g_i|^2 \in \mathbb{R}. \quad (6.43)$$

Unlike in the previous case, the 2-norm is differentiable in all its points. We follow the same procedure that for the analysis of the structure of the 2-norm.

The goal is to minimize the MSE cost function, given by (6.38), subject to the linear model (6.12). We make the same suppositions that in the previous section. To simplify the notation, the variances σ_i^2 are the same for all the extended sources, without loss of generality. The optimization problem now results

$$\min_{g'_1, g'_2} \left\{ E \left\{ |e(k)|^2 \right\} \right\} = \min_{g'_1, g'_2} \left\{ |g'_1|^2 \sigma_i^2 + |g'_2|^2 \sigma_i^2 \right\}, \quad (6.44)$$

$$\text{s.t. } e(k) = (g'_1)^* s'_1(k) + (g'_2)^* s'_2(k). \quad (6.45)$$

The constraint (6.45) is the same that for the previous analysis, so represents an affine function on the plane (g'_1, g'_2) . However, the level curves defined by

$$\frac{E \left\{ |e(k)|^2 \right\}}{\sigma_i^2} = |g'_1|^2 + |g'_2|^2, \quad (6.46)$$

are now circles on the plane (g'_1, g'_2) .

In Figure 6.5, a representation of this plane is shown. The dashed line represents the constraint (6.45). The solution to this problem is $(g'_1, g'_2)_{opt}$, and must be a point over this line. The family of circles on the plane is the geometry locus of the points that satisfy (6.46). The radius of these circles is defined by the value of $E \left\{ |e(k)|^2 \right\} / \sigma_i^2$.

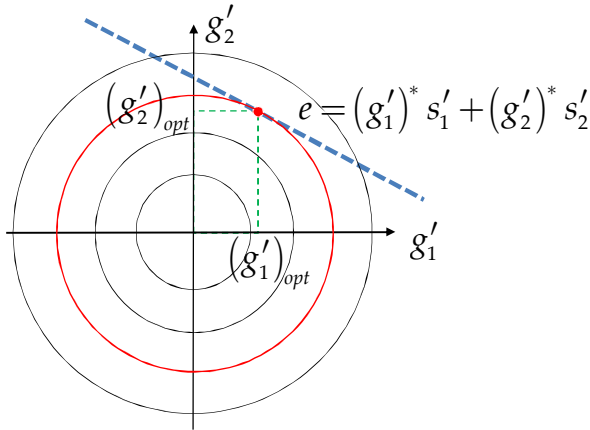


Figure 6.5 - This figure shows the geometry of the 2-norm discussed in this section. The surfaces of equally norm are circles. The axes are the coefficients g'_1 and g'_2 .

The solution to the optimization problem is marked by a red dot, and is given by the point of the dashed line that belongs to the circle of minimum radius. No other point of the constraint line is nearer to the origin, in Euclidean distance.

With this figure, we can see why the solution given by the minimization of the 2-norm has not sparse coefficients. The solution point \mathbf{g}'_{opt} has no zero coefficients because it is not located on one of the coordinate axis.

6.7.3 Geometry of the norms in under-determined problems

As we presented before in this Section, the understanding of the geometry of the norms is crucial to study the solutions given by minimum perimeter of the training error and the MMSE criteria. At this point, we aim to do a similar graphical interpretation of Section 6.7.1 and 6.7.2 when the problem is under-determined.

Suppose a 3-dimensional extension of the study of the norms done before. The cost function

$$\frac{L(e(k))}{L_i} = |g'_1| + |g'_2| + |g'_3| \quad (6.47)$$

defines a family of concentric octahedrons, in a 3D space whose coordinate axes are defined by the coefficients of the extended vector of coefficients. This is

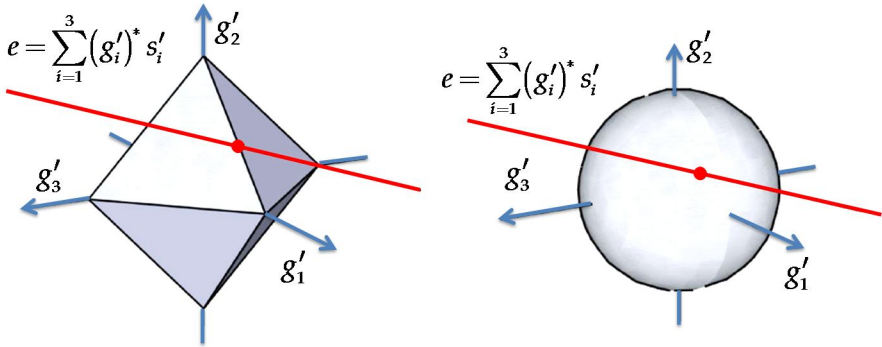


Figure 6.6 – Geometry of the discussion presented in Section 6.7.3. This time, the axes are the coefficients g'_1 , g'_2 , and g'_3 . (left) Geometry of the 1-norm in under-determined problems. (right) Geometry of the 2-norm in under-determined problems.

shown in Figure 6.6 (left) for a certain octahedron, which is supposed to be the one that minimizes the cost function. The subspace defined by

$$e(k) = (g'_1)^* s'_1(k) + (g'_2)^* s'_2(k) + (g'_3)^* s'_3(k) \quad (6.48)$$

is the dashed line. The dimension of this subspace is the degree of freedom available in the system. Observe how at the contact point of the octahedron and the subspace, only one of the three coordinates is zero (g'_3). This means that the solution $(g'_1, g'_2, g'_3)_{opt}$ given by the 1-norm may not have all the coefficients but one set to zero, thus leading to non-exact solutions but cancelling completely some of the interferences.

However, for the MMSE criterion, this is not the case. The cost function

$$\frac{E\left\{|e(k)|^2\right\}}{\sigma_i^2} = |g'_1|^2 + |g'_2|^2 + |g'_3|^2 \quad (6.49)$$

defines a family of spheres in the 3D space. This is also shown in Figure 6.6 (right) for the sphere that corresponds to the minimum of the cost function. Again, the subspace defined by

$$e(k) = (g'_1)^* s'_1(k) + (g'_2)^* s'_2(k) + (g'_3)^* s'_3(k) \quad (6.50)$$

contains the candidates for the solution. The geometry of the 2-norm places the solution $(g'_1, g'_2, g'_3)_{opt}$ in a point where the dashed line touches tangentially the

sphere. In general, this point does not have any coefficients set to zero, so we can conclude that the MMSE criterion does not force the exact cancellation of any of the interferences.

In general, all norms $\|\mathbf{g}\|_p$, with $p > 1$, does not pursue sparse solutions due to the fact that the point of minimum norm is not located on an axis. This discussion explains many of the differences found in Section 6.6 between the minimum convex perimeter criterion and the minimum MSE criterion.

The study of optimization problems that pursues sparse solutions is one of the most active problems in signal processing, with the recent advances in the field of Compressed Sensing. There have been many theoretical and practical advances in the last few years. For more information of this topic, see (Eldar, 2012).

The cost functions presented in this chapter have a noise term that has an influence in the minimization process. This leads to a structural problem that we aim to solve in the next section.

6.8 A Partial Zero-Forcing Criterion

The presence of noise in the cost function (6.17) leads to the following discussion. We may think about minimizing directly this convex perimeter, but there is a high risk of cancelling the second term if the noise is strong enough. This would mean that the extraction vector \mathbf{b} goes to zero and, therefore, the estimated signal (6.5) is null. This is not what we want, so we have to force that \mathbf{b} will not go to zero.

6.8.1 Constraints on the optimization problem

Therefore, it is a good idea to modify this minimum convex perimeter criterion to enforce the joint minimization of the filtered noise while cancelling the number of signals that interfere with our desired source. This is usually known as a *Partial Zero-Forcing (PZF)* behavior, because it tries to cancel as much coefficients of the residual transfer vector \mathbf{g} as possible, but keeping strong the coefficient corresponding to the desired source. See (Johnson Jr, 1995) for an introduction to Zero Forcing.

This PZF-like solution is obtained by the constrained optimization problem

$$\mathbf{b}_0 = \arg \left\{ \min_{\mathbf{b}} \left\{ L(e(k)) \right\} \right\}, \quad (6.51)$$

$$\text{s.t. } |\mathbf{b}^H \hat{\mathbf{a}}_1| = 1, \quad (6.52)$$

where $\hat{\mathbf{a}}_1$ is an estimate of the column of the mixture matrix \mathbf{A} that corresponds with the desired signal. The simplest estimate is given by the solution of the MMSE criterion¹⁷, but other good estimates can be used instead.

Let us take a look on the dimensions involved here. The proposed PZF-like criterion is able to cancel the contribution of $m-1$ of the interference signals and, simultaneously, to preserve the desired source in $y(k)$. Note that the system has only m degrees of freedom given by the sensors, so unfortunately there will be $n-m$ remaining sources that cannot be cancelled by using linear techniques.

The $m-1$ cancelled sources will be the strongest ones, depending on the criterion used. In this case, they are the ones with the largest convex perimeter L_i .

The same analysis can be done for the MSE cost function, yielding to the optimization problem:

$$\mathbf{b}_0 = \arg \left\{ \min_{\mathbf{b}} \left\{ E \left\{ \left| e(k) \right|^2 \right\} \right\} \right\}, \quad (6.53)$$

$$\text{s.t. } |\mathbf{b}^H \hat{\mathbf{a}}_1| = 1. \quad (6.54)$$

However, as stated before, the PZF-like solution to this problem does not guarantee the exact cancellation of any of the sources. This is only due to the characteristics of the 2-norm presented in the MMSE criterion.

6.8.2 The exact Partial Zero-Forcing solution

In order to set a theoretical reference and upper-bound in the performance of the proposed method with PZF behavior, by using the exact knowledge of the mixing matrix \mathbf{A} we compute the *exact PZF solution* that is closest to the one obtained optimizing (6.51)-(6.52). A natural measure of proximity between the PZF solutions is the Euclidean distance in the transformed domain of the coefficients of the residual mixture.

In a situation of sufficient *high Signal to Noise Ratio (SNR)*, the strongest interference does not come from the noise, but from the undesired sources. Thus, we identify the indexes π_1, \dots, π_{m-1} of the $m-1$ sources that the PZF

¹⁷ This estimate is given by the quotient between spatial correlations $\hat{\mathbf{a}}_1 = E\{\mathbf{x}\mathbf{s}_1^*\} / E\{|s_1|^2\}$.

criterion tries to cancel out. These are the sources with the strongest contribution to the output $y(k) = \mathbf{b}_0^H \mathbf{z}(k)$.

With the help of the estimate of the spatial signature of the desired source $\hat{\mathbf{a}}_1$, we form the matrix

$$\mathbf{A}_d = \left[\hat{\mathbf{a}}_1, \mathbf{a}_{\pi_1}, \dots, \mathbf{a}_{\pi_{m-1}} \right], \quad (6.55)$$

where $\mathbf{a}_{\pi_1}, \dots, \mathbf{a}_{\pi_{m-1}}$ are the columns of \mathbf{A} corresponding to these indexes.

The desired exact PZF solution is defined by the following minimum distance problem under linear constraints

$$\mathbf{b}_E = \arg \left\{ \min_{\mathbf{b}} \left\{ \left\| \mathbf{A}^H (\mathbf{b} - \mathbf{b}_0) \right\|_2 \right\} \right\}, \quad (6.56)$$

$$\text{s.t. } \mathbf{A}_d^H \mathbf{b} = \boldsymbol{\delta}_1. \quad (6.57)$$

Theorem 6.2 (The exact Partial Zero-Forcing solution)

Given the optimization problem (6.56)-(6.57), the extraction vector that gives the exact PZF solution closest (in the space of the coefficients of the residual mixture) to \mathbf{b}_0 is

$$\mathbf{b}_E = \mathbf{b}_0 + \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \left(\mathbf{A}_d^H \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \right)^{-1} \left(\boldsymbol{\delta}_1 - \mathbf{A}_d^H \mathbf{b}_0 \right), \quad (6.58)$$

where \mathbf{b}_0 is the solution to the minimum convex perimeter criterion with PZF, given by the optimization of (6.51)-(6.52), and \mathbf{A}_d is the mixture matrix with sorted columns given by (6.55).

Proof: With the help of *Lagrange multipliers* (see (Bertsekas, 1999)), we form the following function that represents the unconstrained optimization problem from (6.56)-(6.57),

$$F(\mathbf{b}) = \underbrace{\left(\mathbf{b}^H - \mathbf{b}_0^H \right) \mathbf{A} \mathbf{A}^H}_{\Delta \mathbf{b}^H} \underbrace{\left(\mathbf{b} - \mathbf{b}_0 \right)}_{\Delta \mathbf{b}} + \left(\boldsymbol{\delta}_1^H - \mathbf{b}^H \mathbf{A}_d \right) \boldsymbol{\lambda}. \quad (6.59)$$

To find the solution \mathbf{b}_E , we minimize this cost function with respect to the parameters. First, the conjugate gradient related to the extraction vector,

$$\mathbf{0} = \nabla_{\mathbf{b}} F(\mathbf{b}) \Big|_{\mathbf{b}=\mathbf{b}_E} = \mathbf{A} \mathbf{A}^H \Delta \mathbf{b}_E - \boldsymbol{\lambda} \mathbf{A}_d, \quad (6.60)$$

obtaining the relation

$$\Delta \mathbf{b}_E = \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \boldsymbol{\lambda}. \quad (6.61)$$

On the other hand, by minimizing with respect to the Lagrange multipliers and by using $\mathbf{b}_E = \Delta \mathbf{b}_E + \mathbf{b}_0$, we obtain:

$$0 = \left. \frac{\partial F(\mathbf{b})}{\partial \boldsymbol{\lambda}} \right|_{\mathbf{b}=\mathbf{b}_E} = \boldsymbol{\delta}_1^H - \mathbf{b}_E^H \mathbf{A}_d = \boldsymbol{\delta}_1^H - \Delta \mathbf{b}_E^H \mathbf{A}_d - \mathbf{b}_0^H \mathbf{A}_d. \quad (6.62)$$

Using the Hermitian of (6.61) in (6.62), one gets the relation

$$\boldsymbol{\delta}_1^H - \mathbf{b}_0^H \mathbf{A}_d = \boldsymbol{\lambda}^H \mathbf{A}_d^H \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d. \quad (6.63)$$

By solving for $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda} = \left(\mathbf{A}_d^H \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \right)^{-1} \left(\boldsymbol{\delta}_1 - \mathbf{A}_d^H \mathbf{b}_0 \right). \quad (6.64)$$

Finally, by substituting (6.64) in (6.61), and by adding \mathbf{b}_0 , we obtain the exact PZF solution,

$$\mathbf{b}_E = \mathbf{b}_0 + \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \left(\mathbf{A}_d^H \left(\mathbf{A} \mathbf{A}^H \right)^{-1} \mathbf{A}_d \right)^{-1} \left(\boldsymbol{\delta}_1 - \mathbf{A}_d^H \mathbf{b}_0 \right). \quad (6.65)$$

■

6.8.3 The gradient of the convex perimeter with PZF

As we have changed the optimization problem to prevent the null values on the extraction vector, the gradient of the cost function must be adapted to this new situation. For that, we first reformulate the constrained optimization problem to have the structure of an unconstrained optimization problem. Thus, the problem (6.51)-(6.52) can be seen as

$$\mathbf{b}_0 = \arg \left\{ \min_{\mathbf{b}} \{ \psi(\mathbf{b}) \} \right\}, \quad (6.66)$$

where the cost function now includes the constraint within:

$$\psi(\mathbf{b}) = L \left[d(k) - \frac{\mathbf{b}^H \mathbf{x}(k)}{|\mathbf{b}^H \hat{\mathbf{a}}_1|} \right]. \quad (6.67)$$

The gradient of the convex perimeter with Partial Zero-Forcing is presented in

the following Theorem.

Theorem 6.3 (The gradient of the convex perimeter with PZF)

Given the cost function (6.67), the gradient needed for the update of the steepest descent algorithm is computed as

$$\nabla_{\mathbf{b}^*} \psi(\mathbf{b}) = \frac{1}{|\mathbf{b}^H \hat{\mathbf{a}}_1|} \left(-\sum_{i=1}^V \frac{\Delta e_i^* \Delta \mathbf{x}_i}{2|\Delta e_i|} + \frac{\hat{\mathbf{a}}_1}{\mathbf{b}^H \hat{\mathbf{a}}_1} \sum_{i=1}^V \frac{\text{Re}\{\Delta e_i^* (\mathbf{b}^H \Delta \mathbf{x}_i)\}}{2|\Delta e_i|} \right), \quad (6.68)$$

where $\hat{\mathbf{a}}_1$ is an estimate of the first column of the mixing matrix.

Proof: We start with the following expression for the error sequence:

$$e(k) = d(k) - y(k) = d(k) - \frac{\mathbf{b}^H \mathbf{x}(k)}{|\mathbf{b}^H \hat{\mathbf{a}}_1|}. \quad (6.69)$$

The gradient of the cost function is

$$\nabla_{\mathbf{b}^*} \psi(\mathbf{b}) = \nabla_{\mathbf{b}^*} L(e) = \sum_{i=1}^V \nabla_{\mathbf{b}^*} |\Delta e_i|. \quad (6.70)$$

The gradients of the differences Δe_i and Δe_i^* are

$$\nabla_{\mathbf{b}^*} (\Delta e_i) = \nabla_{\mathbf{b}^*} \left(\frac{-\mathbf{b}^H \Delta \mathbf{x}_i}{\left((\mathbf{b}^H \hat{\mathbf{a}}_1)(\mathbf{b}^T \hat{\mathbf{a}}_1^*) \right)^{\frac{1}{2}}} \right) = \frac{-\Delta \mathbf{x}_i}{|\mathbf{b}^H \hat{\mathbf{a}}_1|} + \frac{(\mathbf{b}^H \Delta \mathbf{x}_i)}{|\mathbf{b}^H \hat{\mathbf{a}}_1|^2} \nabla_{\mathbf{b}^*} (|\mathbf{b}^H \hat{\mathbf{a}}_1|), \quad (6.71)$$

$$\nabla_{\mathbf{b}^*} (\Delta e_i^*) = \nabla_{\mathbf{b}^*} \left(\frac{-\mathbf{b}^T \Delta \mathbf{x}_i^*}{\left((\mathbf{b}^H \hat{\mathbf{a}}_1)(\mathbf{b}^T \hat{\mathbf{a}}_1^*) \right)^{\frac{1}{2}}} \right) = \frac{(\mathbf{b}^T \Delta \mathbf{x}_i^*)}{|\mathbf{b}^H \hat{\mathbf{a}}_1|^2} \nabla_{\mathbf{b}^*} (|\mathbf{b}^H \hat{\mathbf{a}}_1|), \quad (6.72)$$

with

$$\nabla_{\mathbf{b}^*} (|\mathbf{b}^H \hat{\mathbf{a}}_1|) = \frac{\hat{\mathbf{a}}_1 (\mathbf{b}^T \hat{\mathbf{a}}_1^*)}{2|\mathbf{b}^H \hat{\mathbf{a}}_1|}. \quad (6.73)$$

By substituting (6.73) into (6.71) and (6.72),

$$\nabla_{\mathbf{b}^*} (\Delta e_i) = \frac{1}{|\mathbf{b}^H \hat{\mathbf{a}}_1|} \left(-\Delta \mathbf{x}_i + \frac{(\mathbf{b}^H \Delta \mathbf{x}_i) \hat{\mathbf{a}}_1}{2(\mathbf{b}^H \hat{\mathbf{a}}_1)} \right), \quad (6.74)$$

$$\nabla_{\mathbf{b}^*} (\Delta e_i^*) = \frac{1}{|\mathbf{b}^H \hat{\mathbf{a}}_1|} \left(\frac{(\mathbf{b}^T \Delta \mathbf{x}_i^*) \hat{\mathbf{a}}_1}{2(\mathbf{b}^H \hat{\mathbf{a}}_1)} \right). \quad (6.75)$$

The chain rule on the moduli $|\Delta e_i|$ produces

$$\begin{aligned} \nabla_{\mathbf{b}^*} |\Delta e_i| &= \nabla_{\Delta e_i^*} (|\Delta e_i|) \nabla_{\mathbf{b}^*} (\Delta e_i^*) + \nabla_{\Delta e_i} (|\Delta e_i|) \nabla_{\mathbf{b}^*} (\Delta e_i) \\ &= \frac{1}{2|\Delta e_i| |\mathbf{b}^H \hat{\mathbf{a}}_1|} \left(-\Delta e_i^* \Delta \mathbf{x}_i + \frac{\hat{\mathbf{a}}_1 \left(\Delta e_i (\mathbf{b}^T \Delta \mathbf{x}_i^*) + \Delta e_i^* (\mathbf{b}^H \Delta \mathbf{x}_i) \right)}{2(\mathbf{b}^H \hat{\mathbf{a}}_1)} \right) \\ &= \frac{1}{2|\Delta e_i| |\mathbf{b}^H \hat{\mathbf{a}}_1|} \left(-\Delta e_i^* \Delta \mathbf{x}_i + \frac{\hat{\mathbf{a}}_1 \operatorname{Re} \left\{ \Delta e_i^* (\mathbf{b}^H \Delta \mathbf{x}_i) \right\}}{(\mathbf{b}^H \hat{\mathbf{a}}_1)} \right), \end{aligned} \quad (6.76)$$

where the following gradients have been used:

$$\nabla_{\Delta e_i^*} (|\Delta e_i|) = \Delta e_i^* / 2|\Delta e_i|, \quad (6.77)$$

$$\nabla_{\Delta e_i} (|\Delta e_i|) = \Delta e_i / 2|\Delta e_i|. \quad (6.78)$$

Finally, the summation of all the gradients (6.76) results in (6.68). ■

This new gradient can be directly substituted into the algorithm presented in Table 6.1, to obtain the supervised BCA extraction algorithm with Partial Zero Forcing.

6.9 Computer Simulations

In this section, we aim to illustrate the majority of the concepts presented before. For that, we run several computer simulations varying different parameters.

6.9.1 Signal extraction in an under-determined mixture

Our first experiment is the extraction of the symbols of a desired QPSK source from a linear under-determined mixture. In this example, we consider n sources interfering, but only m sensors. All the information that we have

available consists in T pilot symbols of our source of interest. In general, the length of the training sequence grows with the number of sources, their cardinality and the lack of degrees of freedom in the problem.

The symbols of the interfering sources are totally unknown, and we only know that their supports are bounded (in this case, they are QPSK signals too). The mixture is represented by an unknown matrix \mathbf{A} whose coefficients are drawn from a Gaussian probability density function.

Before reaching each one of the m sensors, the signals are contaminated by a complex AWG noise of zero mean and covariance matrix $\sigma_q^2 \mathbf{I}$. Although this kind of noise is not of finite support and, therefore, it does not satisfy the given assumptions, the obtained results are still reasonable and improve with the increase of the SNR, which is defined by:

$$SNR_{dB} = 10 \log_{10} \left(\frac{\|\mathbf{a}_1\|^2 \sigma_{s_1}^2}{m \sigma_q^2} \right), \quad (6.79)$$

where $\sigma_{s_1}^2$ is the power of the desired signal.

We make use of a gradient descent algorithm where the sample convex perimeter of the error sequence is minimized, step by step, until convergence. After each step, the resulting extraction vector is normalized to force the condition (6.52).

The extended vector of coefficients

We compare three different approaches to the extraction problem. As the system is under-determined, the extraction without residual interferences is not possible and all we can hope is to minimize the number of signals that interferes with our desired source. This means that the extended vector of coefficients \mathbf{g}' should be sparse, as discussed in Section 6.7.3. The three following approaches have been studied in the previous sections:

- The MMSE criterion with PZF, solution to the minimization problem given by (6.53) and (6.54).
- The exact PZF solution, derived in Section 6.8.2, and condensed in equation (6.58). It is used only as a reference because it needs the exact knowledge of the mixing matrix \mathbf{A} , and that information is not available in our problem.
- The criterion of minimum convex perimeter with PZF, proposed in this

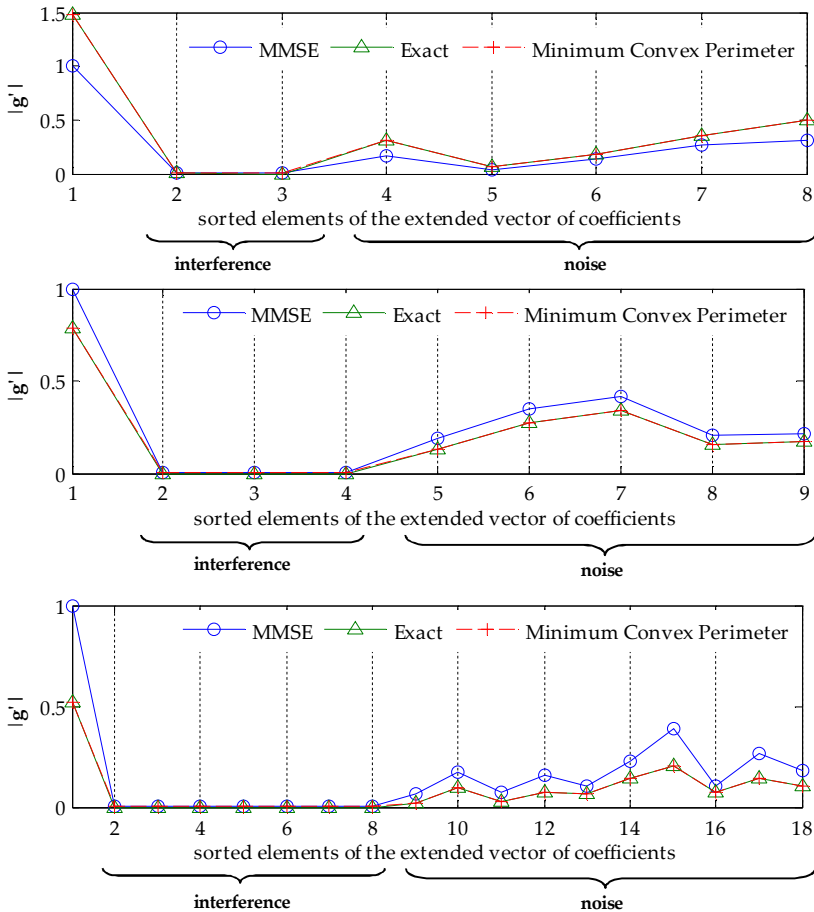


Figure 6.7 – This figure shows the absolute value of the vector of extended coefficients, for three examples of over-determined systems. (upper) $n=3$ sources and $m=5$ observations. (middle) $n=4$ sources and $m=5$ observations. (lower) $n=8$ sources and $m=10$ observations. The first n coefficients correspond to the sources (1 desired signal and $n-1$ interference signals), and the last m coefficients correspond to noise. The exact and the minimum convex perimeter solutions are not normalized.

chapter and given by the solution of (6.51)-(6.52).

We raise progressively the complexity of the mixing model, starting by over-determined systems ($n < m$) and a training sequence of $T = 500$ symbols,

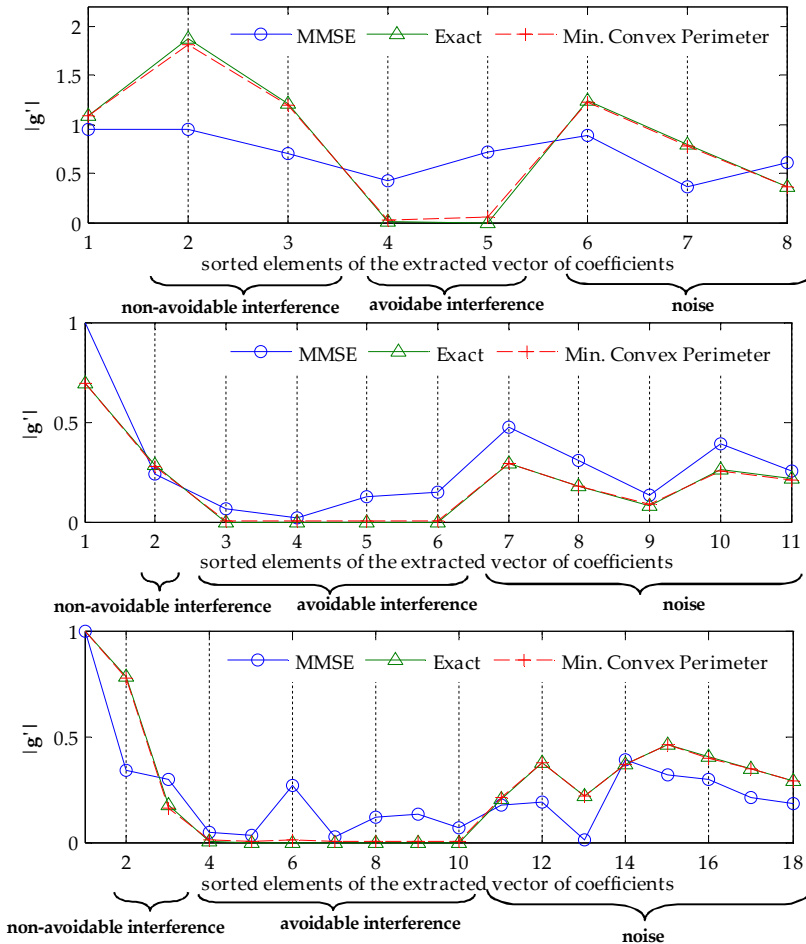


Figure 6.8 – This figure represents the vector of extended coefficients for three examples of under-determined systems. (upper) $n=5$ sources and $m=3$ observations. (middle) $n=6$ sources and $m=5$ observations. (lower) $n=10$ sources and $m=8$ observations. The coefficients are sorted in these plots with the following structure: The first coefficient corresponds to the extracted source, the next $n-m$ coefficients correspond to the non-avoidable interference. The next $n-1$ coefficients correspond to the interference that can be cancelled, due to the available degrees of freedom, while the last m coefficients correspond to m noise signals. The exact and the minimum convex perimeter solutions are not normalized.

where the proposed solution does not improve the MMSE solution (they both cancel the same number of interfering sources). The Figure 6.7 shows the $n + m$ elements of the magnitude of the extended vector of coefficients. The first n coefficients correspond to the sources, and the last m coefficients correspond to noise. A SNR of 30 dB is used. As there are more sensors than sources, all the three options achieve the exact cancellation of the interferences. The use of determined mixture exhibits a similar response.

The next experiment uses under-determined mixtures, so we may expect an altered response from the three different solutions. The Figure 6.8 shows the $n + m$ elements of the magnitude of the extended vector of coefficients. A SNR of 30 dB is used again, so the strongest sources of interference do not come from the noise but from the undesired sources.

However, this time there are more sources than sensors, so the different approaches behave different when they try to cancel out the interference. It can be seen that the proposed solution is quite close to the exact-PZF solution, and contrary to the MMSE-PZF, is able to cancel out almost completely the $m - n$ strongest sources of interference.

This latter approach clearly does not enforce a sparse behavior and all the $n - 1$ sources of interference remains attenuated. We remark that the exact PZF solution is only presented for comparison and is not feasible in practice because it uses the exact knowledge of the mixing matrix \mathbf{A} .

The next concept is a measure on how this sparse interference cancellation is achieved for each of the solutions.

The Performance Index

Let π be a permutation of the indices $2, \dots, n$, which one can use to sort the coefficients of the global transfer vector. We define the following *Performance Index (PI)*, a modification of the Amari Index (Amari, 1996) used in the simulations of Chapter 4, which takes into account the lack of degrees of freedom to cancel all the sources of interference in under-determined mixtures.

Definition (Performance Index (PI)): is defined from the magnitudes of the residual transfer vector.

$$PI(\mathbf{b}) = \frac{1}{m-1} \min_{\pi} \left\{ \sum_{i=1}^{m-1} \frac{|\mathcal{G}_{\pi(i)}|}{|\mathcal{G}_1|} \right\}. \quad (6.80)$$

The smaller this index, the better the cancellation of the interference. It can be

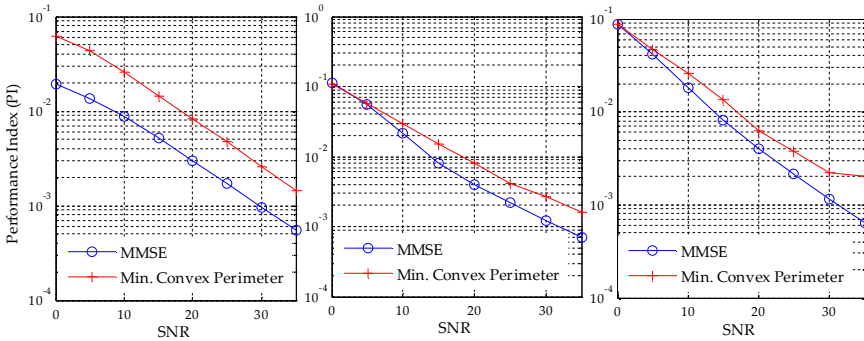


Figure 6.9 - This figure represents the Performance Index (PI) versus the SNR, for the three examples of over-determined systems presented before. (left) $n=3$ sources and $m=5$ observations. (center) $n=4$ sources and $m=5$ observations. (right) $n=8$ sources and $m=10$ observations. The curve corresponding to the exact PZF solution is not shown because it is always zero, since it uses the perfect knowledge of the mixing matrix.

seen as a metric for the successful extraction in under-determined mixture.

The following example illustrates this concept.

Example 6.1 Suppose $n = 4$ sources, $m = 3$ observations, and the following global transference vector at the end of the algorithm,

$$\mathbf{g} = [1 \quad 0.0001 \quad 0.6 \quad 0.0001]. \quad (6.81)$$

The perfect extraction of the first source has not been possible, and there is still a spurious component related to the third source. The possible permutations are,

$$\begin{aligned} \pi_a &= [2 \quad 3 \quad 4]; \pi_b = [2 \quad 4 \quad 3]; \pi_c = [4 \quad 3 \quad 2]; \\ \pi_d &= [4 \quad 2 \quad 3]; \pi_e = [3 \quad 2 \quad 4]; \pi_f = [3 \quad 4 \quad 2]; \end{aligned} \quad (6.82)$$

The corresponding summations in the Performance Index (PI) are $g_{\pi(1)} + g_{\pi(2)}$, and the minimum occurs when the third coefficient is the last one in π . That is, for the permutations π_b or π_d . Hence, the minimum summation is

$$\min_{\pi} \{g_{\pi(1)} + g_{\pi(2)}\} = g_2 + g_4 = 0.0002, \quad (6.83)$$

which produces the Performance Index $PI(\mathbf{b}) = 0.0001$.

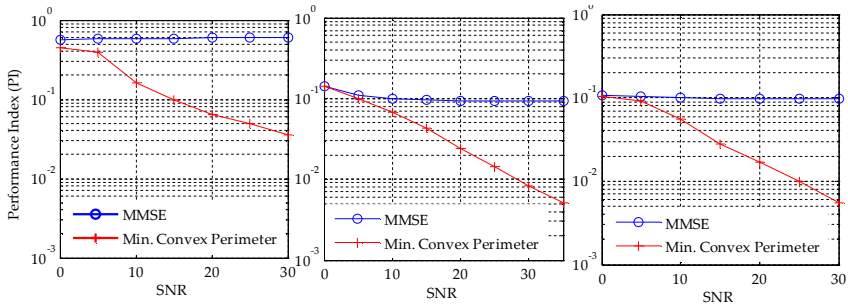


Figure 6.10 - This figure represents the Performance Index (PI) versus the SNR, for the three examples of under-determined systems presented before. (left) $n=5$ sources and $m=3$ observations. (center) $n=6$ sources and $m=5$ observations. (right) $n=10$ sources and $m=8$ observations. The curve corresponding to the exact PZF solution is not shown because it is always zero, since it uses the perfect knowledge of the mixing matrix.

Intuitively, the PI accounts for the magnitudes of the coefficients in \mathbf{g}' that correspond to the $n-1$ sources of interference that can be cancelled (see the avoidable interference Figure 6.8).

In the Figure 6.9, the Performance Index versus the SNR on three different over-determined scenarios is shown. As there are more sources than sensors, the PI is only limited by the presence of noise, and eventually reaches zero in a noiseless situation.

The next experiment uses under-determined mixtures. In Figure 6.10, the Performance Index for three different systems is shown. One can observe that the proposed solution exhibits a better cancellation of the interference than the MMSE. This improvement also increases with high SNR, behavior that will be also observed in further simulations.

The output signal

One can examine the shape of the output signal $y(k)$ in the complex plane. Ideally, for over-determined and determined mixtures, this output should be similar to the original source (in this case, a QPSK constellation). However, in under-determined systems, the presence of spurious interference in the output signal produces the dispersion of the symbols on the complex plane, like is shown in the Figure 6.11 for three different solutions with $SNR = 20$ and $T = 2000$ samples.

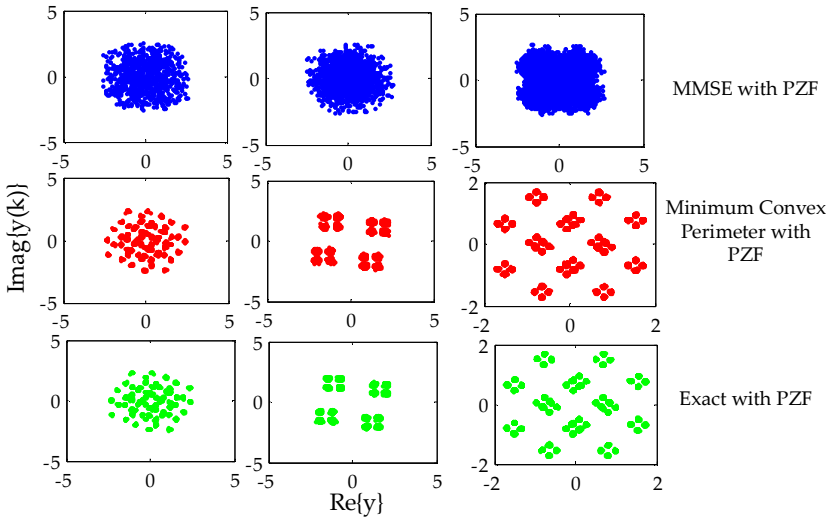


Figure 6.11 – This figure shows the outputs of the extraction system in the complex plane, for three different under-determined mixtures. (left column) $n=5$ sources and $m=3$ observations. (center column) $n=6$ sources and $m=5$ observations. (right column) $n=10$ sources and $m=8$ observations. The first row (blue dots) corresponds to the MMSE solution, the second row (red dots) corresponds to the minimum convex perimeter solution, and the third row (green dots) corresponds to the exact solution.

However, it can be seen that the BCA of the training error has condensed most of the relevant information of the observations in a linear estimate of the desired signal, which can be later used for subsequent refine estimation. It is clear from the figure that the proposed PZF criterion groups the samples in few clusters than the MMSE-PZF criterion, whose symbols are scattered throughout the complex plane.

This fact can be later exploited by non-linear techniques to perform an improved classification of the transmitted symbols.

6.9.2 Use as a pre-processing step for classification

We can exploit the property that the residual transfer vector \mathbf{g} is sparse in the sense of the number of active components, to simplify the *decisor* of a

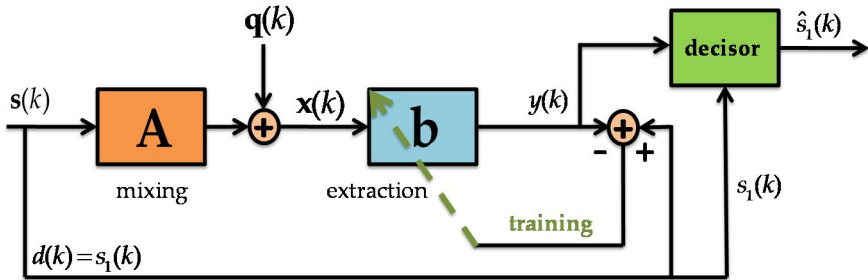


Figure 6.12 – This figure shows the block diagram of the complete system of Section 6.9.2. After the extraction, the output is passed through a decisor which has been trained with the same pilot symbols than the extraction vector. The result is a non-linear estimation of the original source.

communications receiver, which basically implements a *classifier*¹⁸.

It is well known that non-linear estimates, like those based on the *Maximum Likelihood (ML)* and *Maximum A Posteriori (MAP)* criteria (Kay, 1993), can outperform the accuracy of linear estimates like the supervised BCA extraction algorithm. However, as we do not have other prior knowledge of the sources of interference than their bounded support, these computations are no longer feasible.

Therefore, we adopt an alternative strategy which leads to a convenient solution of the problem. It consists on using first, as a preprocessing step, a linear estimate that summarizes the relevant information of the observations by preserving the signal of interest while cancelling most of the sources of interference.

This is what we have been doing during this entire chapter. After that, a classifier can recover the original symbols from the desired source if the input is enough clean of interferences.

In the Figure 6.12, a block diagram of the complete system is shown. It consists on the addition of a classifier at the output of the extraction system shown before, so the estimated source can be further refined by non-linear techniques. Two different solutions have been tested to solve this problem.

¹⁸ A classifier assigns a class to each one of the extracted symbols, based on a certain criterion.

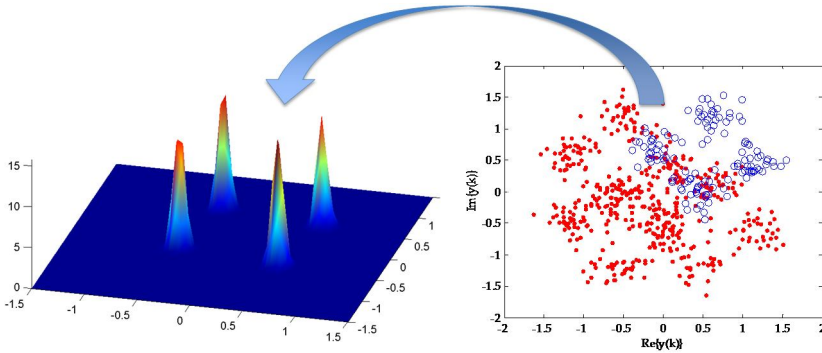


Figure 6.13 – The left figure shows the construction of the multimodal p.d.f. used in the MAP decisor for the symbol $1 + j$. In the right figure, the output of a BCA extraction algorithm is shown, for an under-determined system with $n = 6$ and $m = 5$. The blue circles are the samples of $y(k)$ for the indices k that satisfy $s_1(k) = 1 + j$.

Implementing a hard decisor based on the Maximum A Posteriori rule

Let Γ denote the alphabet of symbols of the signal of interest. In a QPSK signal with power $\sigma_s^2 = 2$, the alphabet is:

$$\Gamma = \{1 + j, 1 - j, -1 + j, -1 - j\}. \quad (6.84)$$

The output of the decisor is given by the maximization of a probabilistic criterion.

Definition (Maximum A Posteriori (MAP) Rule): The Maximum A Posteriori decisor is implemented by the following rule, symbol by symbol:

$$\hat{s}_1(k) = \arg \max_{s_1(k) \in \Gamma} \{p(s_1(k) | y(k))\}, \quad \text{for } k = 1, \dots, T. \quad (6.85)$$

To implement this in practice, we have to build the contrast $p(s_1(k) | y(k))$ for each one of the possible symbols, $s_1(k) \in \Gamma$. These functions can be seen as mountain landscapes, built from the samples $y(k)$ shown in Figure 6.11.

The process consists on the clustering of the samples that corresponds to a certain symbol, and on the construction of a sum of 35 Gaussian p.d.f.s whose means and variances are computed from the samples themselves.

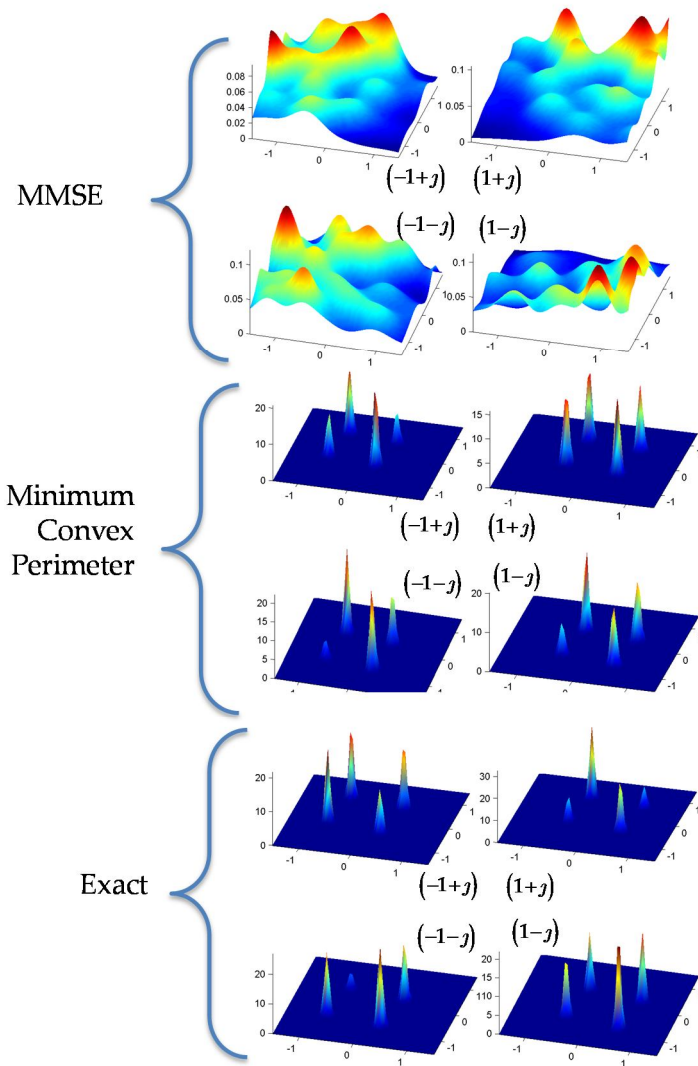


Figure 6.14 – In this figure, the four surfaces built for the MAP decision are shown, for the three different solutions with PZF discussed during the whole chapter: the MMSE solution (upper), the minimum convex perimeter solution (middle) and the exact solution (lower). The under-determined mixture uses $n = 6$ sources and $m = 5$ observations, and the SNR = 15 dB. We took $T = 2000$ symbols for the training stage.

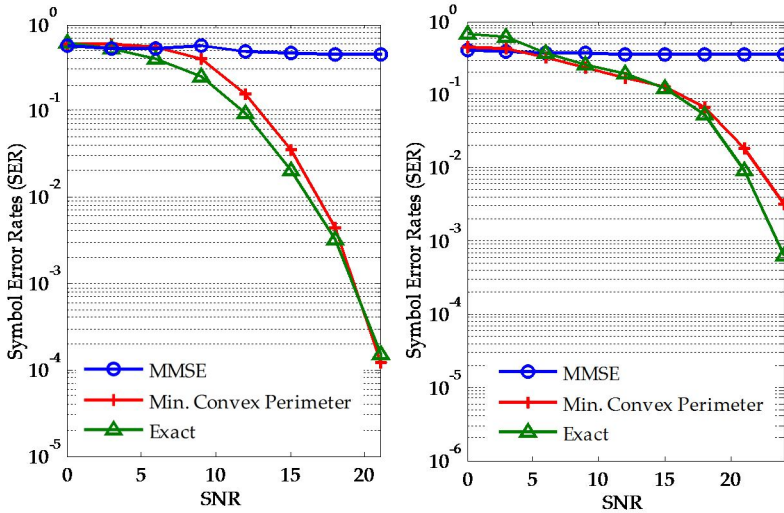


Figure 6.15 – SER curves for the three solutions explored in this chapter, and two different under-determined experiments. (left) $n = 5$ sources and $m = 3$ observations. (right) $n = 7$ sources and $m = 5$ observations. The decision is based on a MAP classifier.

For example, for the symbol $1 + j$, we construct a multimodal p.d.f. with the samples of $y(k)$ that comes from a symbol $1 + j$ in $s_1(k)$. This is shown in Figure 6.13 for an under-determined mixture with $SNR = 15$ dB and a training sequence of $T = 2000$ symbols.

We repeat the process for the other three symbols in Γ , to end the training process of the MAP decisor. After that, we can receive unknown samples and decide that the correct symbol for each sample of $y(k)$ is the one whose p.d.f. value is maximized.

Figure 6.14 shows the contrast function for each symbol on the alphabet Γ . It can be seen that the MAP surfaces synthetized by the minimum convex perimeter and the exact solutions are sparser than the one given by the MMSE solution. Thus, the decision is easier on those surfaces and leads to less errors.

The decision error is usually measured by the *Symbol Error Rate (SER)* curves, which are widely used in all communication standards. We sent $T_{SER} = 10^6$ unknown symbols after the training of the extraction and the decisor subsystems. The SER curves quantify, for each SNR, the ratio between the number of wrong decided points and T_{SER} .

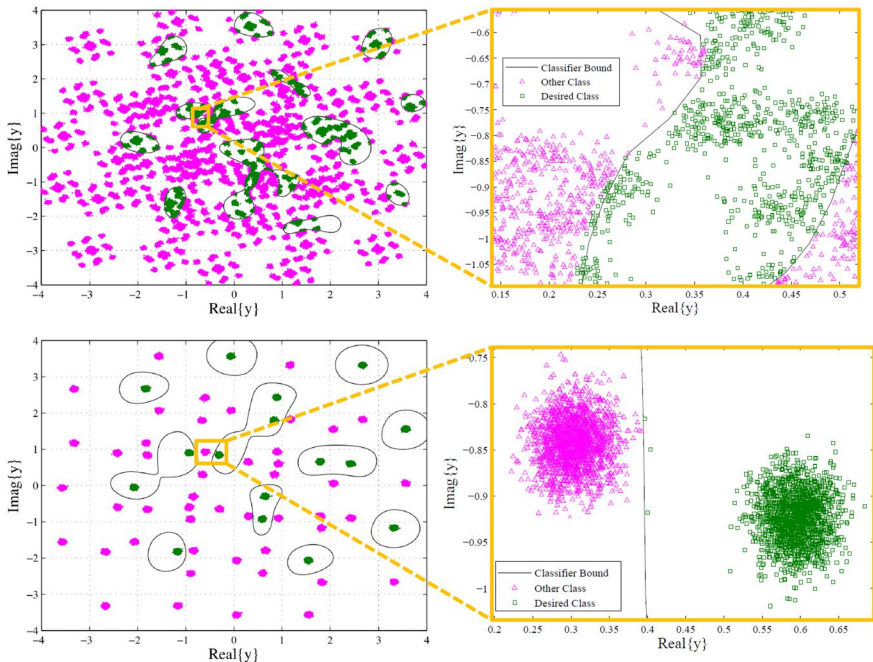


Figure 6.16 - This plot shows the result of a classification of symbols by using a SVM. The green points are samples of the output $y(k)$ that correspond to the input symbol $1 + j$, while the magenta points correspond to the input symbols $1 - j, -1 - j$, and $-1 + j$. The upper row shows the output of the MMSE criterion, while the lower row shows the output of the minimum convex perimeter criterion. The right column zooms into a conflictive region on the complex plane. The decision bounds given by the SVM are shown as black lines rounding some regions.

$$SER = \frac{\# \text{ wrong decided symbols}}{T_{SER}}. \quad (6.86)$$

The Figure 6.15 shows the SER curves for two different scenarios. In this comparison, we again observe how the proposed criterion outperforms the MMSE and stays close to the performance bound of the exact solution (that has been computed from the exact knowledge of the mixing system). Thus, after a short training stage, the minimum convex perimeter criterion is able to recover successfully unknown symbols in under-determined mixtures.

Implementing a Support Vector Machine

Instead of using a MAP classifier with hard decision, one can implement a *Support Vector Machine (SVM)*, which assigns one of the four classes in Γ to each one of the samples in the complex plane received (Cristianini, 2000). The SVM selected is based on a *Radial Basis Function kernel* with a scaling factor of 0'2 (see (Bishop, 2006)).

As in the previous case, the SVM is able to classify unknown symbols of the desired source after a short training process. The Figure 6.16 shows the result of a classification by using the solution of a MMSE criterion and a minimum convex perimeter criterion, both with PZF constraints. The parameters for this simulation were: $n = 5$ sources and $m = 3$ observations, a training length of $T = 600$ symbols and $SNR = 25$ dB .

It can be seen that the sparse behavior of the proposed method groups the points into small portions of the complex plane, while an important number of points are misclassified when the MMSE method is used, due to the fact that the 2-norm does not enforces sparse behaviors.

After the classification, one can count for the number of errors that happen when a sequence of $T_{SER} = 10^6$ symbols is sent. The Figure 6.17 shows that the SVM classifier after the MMSE extracted signal is unable to classify properly the symbols of the desired source, while the minimum convex perimeter solution improves it rate of success with high SNR.

6.10 Conclusions

In this chapter, we presented the Bounded Component Analysis of the training error, a supervised or semi-blind method to extract a bounded source from a linear mixture. The system model can be under-determined and the presence of AWG noise is supposed. After the training stage, further unknown samples of the desired signal can be recovered if the mixture does not change over time. The main lines of the chapter are emphasized here:

- The minimum convex perimeter of the training error provides an appropriate criterion for the minimization of the error sequence, which is the difference between the reference and the estimation.
- The convex perimeter functional shares some similitudes with the Mean Square Error, like the absence of local minima and the formulation of the cost function as a weighted sum of norms. However, both criteria have important differences, like the lack of an analytic

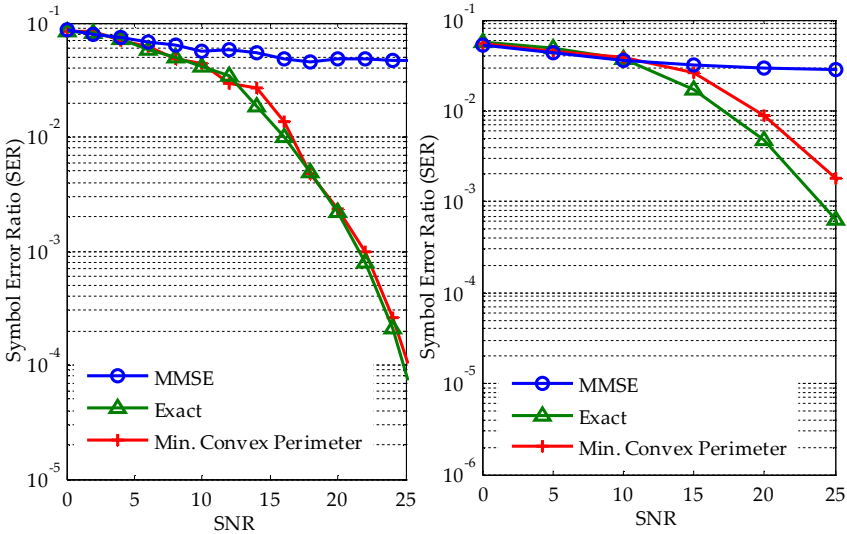


Figure 6.17 – SER curves for the three solutions explored in this chapter, and two different under-determined experiments. (left) $n = 5$ sources and $m = 3$ observations. (right) $n = 6$ sources and $m = 5$ observations. The decisor is based on a SVM classifier.

solution for the optimal value of the extraction vector in the proposed cost function. In addition, the MSE criterion does not force a sparse response on the global transfer vector, in opposition to the convex perimeter.

- The under-determined mixture produces the existence of residual components on the recovered signal. The presence of noise in the model requires the use of a Partial Zero-Forcing approach, where we guarantee that the extraction vector is not driven to zero.
- The presence of spurious components at the output of the algorithm can be solved by a post-processing based on decision criteria like the use of SVM or a MAP rule. Unlike the proposed criterion, the MMSE is not able to prepare the signals to this process.

To illustrate the performance of the minimum convex perimeter, we ran several computer simulations, changing the parameters to conduct a complete study of the behavior of the methods developed in the chapter.

7 CONCLUSIONS

Somewhere, something incredible is waiting to be known.

- C. Sagan -

This Thesis started with the study of the calculus with complex variables. To work in this framework, we adapted some of the definitions to act properly on complex vectors and processes. With Wirtinger calculus, we presented the main tools to deal with the problem of non-holomorphic functions.

These tools are based essentially on the independence between the variables and its conjugate counterparts. For example, it is important to remark that the optimization of a cost function goes on the direction of the gradient with respect to the conjugate vector.

Although the detailed conclusions of this Thesis are summarized at the end of each chapter, in this section we give an overview of the key points. At first, we showed some of the most important applications in the field of complex-valued signal processing. There were both supervised and blind examples, but they do

not work in some special scenarios (dependent sources in BSS, and under-determined mixtures in supervised source extraction), which we aim to solve with the later proposed techniques.

Also, we have presented some recent advances in the research framework of complex-valued signal processing. Specifically, when there are non-circular signals in the problem, the use of augmented statistics and widely linear modeling provides additional information that can outperform the classical techniques.

Three related lines of research have been conducted in the second part of this Thesis, whose main conclusions are outlined below:

- Many signals that are used in communications standards are complex-valued, and exhibit a bounded shape (called constellation) in the complex plane. To exploit this property in the context of signal processing, we studied the theory of Bounded Component Analysis, a novel blind technique for the separation, extraction and decomposition of linear mixtures. The main assumptions are the bounded nature and the Cartesian decomposition of the support of the sources.

To provide the identifiability of the mixture and the separability of the components, we showed the invertibility of the mixture and the Minkowski direct decomposability of the convex hull of the observations. This result can also be applied when the sources are somehow dependent, where techniques like Independent Component Analysis fails.

Experimental simulations with communication signals confirmed the advantages of the proposed criterion in high SNR scenarios. We showed the use of a gradient descent algorithm in three related applications: signal extraction, signal separation and component decomposition.

- When minimizing a cost function via a gradient descent algorithm, it is important to analyze the behavior of the iterations. By doing this, one can obtain valuable information about the location of the minima, the speed of convergence or the reliability of the solutions. For this purpose, we analyzed the convergence of a BCA extraction algorithm.

The proposed contrast function is based on the normalized convex perimeter of the output, and was free of erroneous local minima. We started by proposing fast although potentially unstable updates, based on the Newton-Raphson step size and a step size that uses the second-

order information.

The update equation was studied by using the structure of the gradient. After that, a condition for the global convergence of the gradient descent algorithm was established, with the proposal of a step size that guarantees the stability.

However, it was shown that it depends on parameters that cannot be evaluated in practice, so feasible estimators were proposed too. These step sizes are slightly slower than the ones based upon local approximations of the cost functions, but they guarantee the stability of the algorithm.

- When the model is more complicated than the determined and noiseless system presented in the blind case, the use of a short training sequence can help the algorithm to recover the desired signal, even when there are not enough degrees of freedom. For this purpose, we presented the Bounded Component analysis of the training error, which aims at the linear extraction of one signal of interest in an under-determined mixture.

As in the blind case, the main assumptions were the bounded nature and the Cartesian decomposition of the supports of the sources and of the noise. Also, a Partial Zero-Forcing constraint was introduced to help to the algorithm to succeed in noisy scenarios, where there exists the risk of cancelling the estimated signal if the noise is strong enough.

The minimum convex perimeter criterion shared some relevant structural similitude with the Minimum Mean Square Error criterion, which includes the absence of local minima. However, there was also a key difference in the sense that the proposed estimate prefers those solutions where a small number of sources of interference are active.

The basis of this property is studied from a geometric point of view. We have also shown through simulations that this behavior can be exploited to cancel out the dominant sources of interference, and to perform a subsequent non-linear refined estimation of the signal of interest.

8 OPEN RESEARCH LINES

*If a cluttered desk is a sign of a cluttered mind, of what, then,
is an empty desk a sign?*

- A. Einstein -

We have planned both theoretical and practical extensions of the main contributions of this Thesis. Some of these open research lines have already begun to produce results but needs further refinement, and others are in an earlier stage of study.

We have started to work with high dimensional arrays like tensors, aiming to describe a richer structure on the mixing models (Crespo, 2013). Tensor factorizations have been recently used to solve the problem of blind and supervised signal separation (Cichocki, 2009). The additional dimensions of these objects allow a deeper description of the intricate relations between inputs and outputs of the mixing system, possibly being non-linear and non-stationary. However, the extension of BCA to tensors has not been yet addressed.

Another interesting topic is the adaptation to the widely linear model (Mandic,

2009) of the BCA basic linear model for signal extraction. However, the first approaches to this topic have not produced significant improvements. The main cause is that the direct extension of the model violates the support decomposition of the sources, which is a crucial assumption of BCA. Nevertheless, we need to investigate if a different approach to this issue may circumvent this setback.

The BCA extraction method can be modified to recover a bounded signal from a convolutive channel, thus providing a blind deconvolution scheme for linear filters (Haykin, 1994). Under certain conditions, we suspect that our algorithms can outperform the solutions given by other classical approaches.

We also intend to research further into the convergence of the BCA extraction algorithm, ensuring the full global convergence of the iterations to a local minimum (not only the monotonous convergence). In addition, in order to establish some guides for the value of the step size, the stability study of the semi-blind BCA extraction algorithm must be done.

The actual implementations of the BCA algorithms presented in this Thesis uses a gradient batch approach, where the convex perimeter is computed from scratch at each iteration. This implies the computation of the convex hull by taking into account the whole set of samples of the sequence. Our intention is to modify the gradient descent algorithm to update the parameters by taking into account only the new sample acquired at the present iteration. To do this, we need to review the tools provided by computational geometry (Mount, 2002) for the efficient computation of the convex perimeter.

Finally, maybe the research line with more promising future is the extrapolation of the convex perimeter functional to high dimensions. The recent use of quaternions and Clifford algebra (Via, 2011) in BSS problems suggests that one can propose different functionals apart from the well-known convex perimeter. Algorithms based on quaternions provide a very fast convergence and allow the use of models from the hypercomplex algebra to study real world systems.

REFERENCES

- Abdi, H. and Williams, L.J. 2010.** Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2010.
- Adali, T. and Haykin, S. 2010.** *Adaptive Signal Processing*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- Aguilera, P., Cruces, S, Durán-Díaz, I., and Sarmiento, A. 2013.** Study of the global convergence of a Bounded Component Analysis extraction algorithm. *IEEE Transactions on Signal Processing (under preparation)*. 2013.
- Aguilera, P., Cruces, S., Durán-Díaz, I., Sarmiento, A, and Mandic, D. P. 2012.** Blind separation of dependent sources with a Bounded Component Analysis deflationary algorithm. *IEEE Signal Processing Letters (submitted)*. 2012.
- Aguilera-Bonet, P., Duran-Díaz, I, Sarmiento, A., and Cruces, S. 2012.** Bounded Component Analysis of the training error. *International Joint Conference on Neural Networks (IJCNN)*. Conference Publications, 2012.
- Amari, S.I., Cichocki, A., and Yang, H.H. 1996.** A new learning algorithm for blind source separation. *Advances in Neural Information Processing Systems*. 1996, Vol. 8.
- Apostol, T. M. 1967.** *Calculus, 2nd ed., Vol. 1: One-Variable Calculus, with an Introduction to Linear Algebra*. Waltham, MA: Blaisdell, 1967.
- Arfken, G. 1985.** *Mathematical Methods for Physicists, 3rd ed.* Orlando, FL: Academic Press, 1985.
- Benesty, J., Chen, J., and Huang. Y. 2010.** On widely linear Wiener and tradeoff

- filters for noise reduction. *Journal of Speech Communication*. 2010, Vol. 52, 5.
- Bertsekas, D. P. 1999.** *Nonlinear Programming (Second ed.)*. Cambridge : Athena Scientific, 1999.
- Bingham, E. and Hyvärinen, A. 2000.** A fast fixed-point algorithm for independent component analysis of complex-valued signals. *Int. J. of Neural Systems*. 2000, Vol. 10, 1.
- Bishop, C. M. 2006.** *Pattern Recognition and Machine Learning*. New York : Springer-Verlag, 2006.
- Boyd, S., and Vandenberghe, L. 2009.** *Convex Optimization*. Cambridge : Cambridge University Press, 2009.
- Brandwood, D.H. 1983.** A Complex Gradient Operator and its Application in Adaptive Array Theory. 1983, pp. Vol. 130, No. 1, pp 11-16.
- Cardoso, J. 1997.** Higher order contrasts for independent component analysis. *Neural computations*. 1997, Vol. 11.
- Cardoso, J.F. 1998.** Blind signal separation: statistical principles. *Proceedings of the IEEE*. 1998, Vol. 86, 10.
- Chan, T.M. 1996.** Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete and Computational Geometry*. 1996, Vol. 16.
- Chichocki, A. and Amari, S.-I. 2002.** *Adaptive Blind Signal and Image Processing*. s.l. : John Wiley & Sons, 2002.
- Cichocki, A., Zdunek, R., Phan, A.H. and Amari, S.-I. 2009.** *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. s.l. : Wiley, 2009.
- Comon, P. 2004.** Contrast, independent component analysis, and blind deconvolution. *Int. J. Adapt. Control Signal Processing*. 2004, Vol. 18, 3.
- Comon, P. and Jutten, C. 2010.** *Separation, Handbook of Blind Source*. s.l. : Academic Press, 2010.
- Comon, P. 1994.** Independent Component Analysis: a new concept? *Signal Processing*. 1994, 36(3):287–314.
- Comon, Pierre. 2004.** Blind identification and source separation in 2x3 underdetermined mixtures. *IEEE Transactions on Signal Processing*. 2004, Vol. 52, 1.
- Crespo, C, Aguilera, P., Becerra, J. A., Cruces, S. 2013.** On nonlinear amplifier

modeling and identification using baseband Volterra-Parafac models. *IEEE Signal Processing Letters (submitted)*. 2013.

Cristianini, N. and Shawe-Taylor, J. 2000. *Support Vector Machines and other kernel-based learning methods*. s.l. : Cambridge University Press, 2000.

Croft, H. T., Falconer, K. J., and Guy, R. K. 1991. *Unsolved Problems in Geometry*. New York : Springer-Verlag, 1991.

Cruces, S. A. 2010. Bounded Component Analysis of linear mixtures: A criterion of minimum convex perimeter. *IEEE Transactions on Signal Processing*. 2010, Vol. 58, 4.

Cruces, S. 2008. BCA extraction algorithm in MatLab. <http://personal.us.es/sergio/alg/BCA.html>. [Online] 2008.

Cruces, S., Aguilera-Bonet, P., Durán-Díaz, I. 2011. Criterion for signal extraction in underdetermined mixtures of bounded support. *Signal Processing*. 2011, Vol. 91, 10.

Cruces, S., and Durán, I. 2004. The minimum support criterion for blind signal extraction: a limiting case of the strengthened Young's inequality. *Proceedings of the International Conference on ICA and BSS, Granada, Spain*. 2004.

Cruces, S., Cichocki, A. and De Lathauwer, L. 2004. Thin QR and SVD factorizations for simultaneous blind signal extraction. *proc. of the European Signal Processing Conference (EUSIPCO)*. 2004.

Cruces, S., Cichocki, A., and Amari, S.-I. 2004. From blind signal extraction to blind instantaneous signal separation: criteria, algorithms and stability. *IEEE Transactions on Neural Networks*. 2004, Vol. 15, 4.

Cruces, S., Durán-Díaz, I. Sarmiento, A., and Aguilera, P. 2010. Bounded Component Analysis of Linear Mixtures. *International Conference on Acustics, Speech and Signal Processing (ICASSP)*. Dallas, Texas. 2010.

Damen, M.O., El Gamal, H., and Caire, G. 2003. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions On Information Theory*. 2003, Vol. 49, 10.

Delfosse, N and Loubaton, P. 1995. Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*. 1995, Vol. 45.

Delmas, J. P. 2004. Asymptotically minimum variance second-order estimation for noncircular signals with application to DOA estimation. *IEEE Trans. Signal Processing*. 2004, Vol. 52.

- Diniz, P. 1997.** *Adaptive Filtering: Algorithms and Practical Implementation*. s.l. : Kluwer Academic Publishers, 1997.
- Donoho, D. 2006.** For most large underdetermined systems of linear equations, the minimal 1-norm solutions is also the sparsest solution. *Communications on Pure and Applied Mathematics*. 2006, Vol. 59, 6.
- Durán-Díaz, I., Cruces, S., Sarmiento, A. S., and Aguilera-Bonet, P. 2012.** Solving permutations in frequency-domain for blind separation of an arbitrary number of speech sources. *JASA Express Letters*. 2012, Vol. 131, 2.
- Durán-Díaz, I., Cruces, S., Sarmiento, A., and Aguilera-Bonet, P. 2011.** Cyclic maximization of non-Gaussianity for blind signal extraction of complex-valued sources. *Neurocomputing*. 2011, Vol. 74, 17.
- Elad, M. 2004.** *Sparse Representations of Signals: Theory and Applications*. IPAM – MGA Program. 2004.
- Eldar, Y. C. and Kutyniok, G. 2012.** *Compressed Sensing: Theory and Applications*. s.l. : Cambridge University Press, 2012.
- Erdogan, A. T. 2007.** Globally Convergent Deflationary Instantaneous Blind Source Separation Algorithm for Digital Communications Signals. *IEEE Transactions on Signal Processing*. 2007, Vol. 55, 5.
- Erdogan, A. T. 2006.** A simple geometric blind source separation method for bounded magnitude sources. *IEEE Transactions on Signal Processing*. 2006, Vol. 54, 2.
- Farouki, R.T., Moon, P.H., and Ravani, B. 2001.** Minkowski Geometric Algebra of Complex Sets. *Geometriae Dedicata*. 2001, Vol. 85.
- Gamboa, F. and Gasiat, E. 1997.** Source separation when the input sources are discrete or have constant modulus. *IEEE Transactions on Signal*. 1997, Vol. 45, 12.
- Golub, G., and Van Loan, C. 1996.** *Matrix computations (3rd Edition)*. Baltimore, MD : Johns Hopkins University Press, 1996.
- Górriz, J. M., Ramírez, J., Cruces-Alvarez, S., Puntonet, C. G., Lang, E.W., and Erdogmus, D. 2010.** A Novel LMS algorithm applied to Adaptive Noise Cancellation. *IEEE Signal Processing Letters*. 2010, Vol. 127, 4.
- Graham, R.L. 1972.** An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*. 1972, Vol. 1.
- Gray, R.M., and Davisson, L.D. 2010.** *Introduction to Statistical Signal Processing*. s.l. : Cambridge University Press, 2010.

- Hayes, M. 1996.** *Statistical Digital Signal Processing and Modeling*. s.l. : Wiley, 1996.
- Haykin, S. 1994.** *Blind Deconvolution*. s.l. : Prentice Hall, 1994.
- Herstein, I.N. 1964.** *Topics in Algebra*. Lexington, MA : Xerox College Publishing, 1964.
- Hjørungnes, A. 2011.** *Complex-Valued Matrix Derivatives*. Oslo : Cambridge University Press, 2011.
- Horn, R. and Jhonson, C. 1985.** *Matrix Analysis*. New York : Cambridge University Press, 1985.
- Hotelling, H. 1935.** The most predictable criterion. *J. Ed. Psych.* 1935, Vol. 26.
- Hyvärinen, A. and Oja, E. 1997.** A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*. 1997, Vol. 9, 7.
- Hyvärinen, A., Karhunen, J., and Oja, E. 2001.** *Independent Component Analysis*. s.l. : John Wiley & Sons, 2001.
- Javidi, S., Mandic, D.P., and Cichocki, A. 2010.** Complex Blind Source Extraction From Noisy Mixtures Using Second-Order Statistics. *IEEE Transactions on Circuits and Systems – I: Regular*. 2010, Vol. 57, 7.
- Javidi, S., Pedzisz, M., Goh, S. L., and Mandic, D. P. 2008.** The augmented Complex Least Mean Squares algorithm with application to adaptive prediction problems. *Proceedings of the Cognitive Information Processing Systems Conference (CIP)*. 2008.
- Johnson Jr, C.R. 1995.** On the interaction of adaptive filtering, identification and control. *IEEE Signal Processing Magazine*. March, 1995.
- Jolliffe, I.T. 2002.** *Principal Component Analysis*. New York : Springer Series in Statistics, 2002.
- Kay, S. 1993.** *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. s.l. : Prentice Hall, 1993.
- Koivunen, V. and Eriksson, J. 2006.** Complex Random Vectors and ICA Models: Identifiability, Uniqueness and Separability. *IEEE Transactions on Information Theory*. 2006, Vol. 52, 3.
- Kreutz-Delgado, K. 2006.** *The Complex Gradient Operator and the CR-Calculus*. Sand Diego : University of California, 2006.
- Kreyszig, E. 1988.** *Advanced Engineering Mathematics, 6th ed.* New York : Wiley, 1988.

- Li, Y., Cichocki, A., Amari, S.-I. 2003.** Sparse Component Analysis for Blind Source Separation with Less Sensors than Sources. *4th International Symposium on Independent Component Analysis and Blind Signal Separation*. 2003.
- Liu, W., and Mandic, D.P. 2006.** A normalised kurtosis based algorithm for blind source extraction from noisy measurements. *Signal Processing*. 2006, Vol. 86.
- Looney, D. and Mandic, D.P. 2011.** Augmented complex matrix factorisation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2011.
- Mandic, D. P. and Lee Goh, V. S. 2009.** *Complex Valued Nonlinear Adaptive Filters*. s.l. : John Wiley & Sons, Ltd., 2009.
- Mandic, D.P. and Yamada, I. 2007.** Machine learning and signal processing applications of fixed point theory. *Tutorial in IEEE ICASSP*. 2007.
- Mansour, A., A., Ohnishi, N., and Puntonet, C.G. 2002.** Blind multiuser separation of instantaneous mixture algorithm based on geometrical concepts. *Signal Processing*. 2002, Vol. 82.
- Melnick, E.L., and Tenenbein, A. 1982.** Misspecifications of the Normal Distribution. *The American Statistician*. 1982, Vol. 36, 4.
- Mooers, C. N. 1973.** A technique for the cross spectrum analysis of pairs of complex-valued time series, components and rotational invariants. *Deep-Sea Res.* 1973, Vol. 20.
- Mount, D. M. 2002.** Computational Geometry, Lecture Notes for CMSC 754. *University of Maryland*. 2002.
- Neeser, F., and Massey, J. 1993.** Proper Complex Random Processes with Application to Information Theory. *IEEE Transactions on Information Theory*. 1993, Vol. 39, 4.
- Novoy, M. and Adali, T. 2006.** Adaptable nonlinearity for complex maximization of nongaussianity and a fixed-point algorithm. *Proc. IEEE Workshop on Machine Learning for Signal Processing (MLSP)*. 2006.
- Novoy, M., and Adali, T. 2008.** On extending the complex FastICA algorithm to noncircular sources. *IEEE Trans. Signal Processing*. 2008, Vol. 56, 5.
- Papadias, C. B., 2000.** Globally convergent blind source separation based on a multiuser kurtosis maximization criterion. *IEEE Transactions on Signal Processing*. 2000, Vol. 48.

- Pedzisz, M., and Mandic, D.P. 2007.** *Augmented Complex Statistics for Signal Prediction*. s.l. : Technical Report: TR-ICU-EPSRC-08-07-TR002, 2007.
- Pham, D. T. 2000.** Blind Separation of instantaneous mixture of sources based on order statistics. *IEEE Transactions on Signal Processing*. 2000, Vol. 401, 6755.
- Picinbono, B. 1996.** Second-Order Complex Random Vectors and Normal. *IEEE Transactions on Signal Processing*. 1996, Vol. 44.
- Picinbono, B. 1994.** On circularity. *IEEE Transactions on Signal Processing*. 1994, Vol. 42, 12.
- Picinbono, B., and Chevalier, P. 1995.** Widely linear estimation with complex data. *IEEE Transactions on Signal Processing*. 1995, Vol. 43, 8.
- Preparata, F.P. and Hong, S.J. 1977.** Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*. 1977, Vol. 20, 2.
- Priestley, M.B. 1981.** *Spectral Analysis and Time Series*. s.l. : Academic Press, 1981.
- Qiao, S. 2007.** Fast SVD of Takagi Factorization Package. [Online] 2007. <http://www.cas.mcmaster.ca/~qiao/software/takagi/matlab/readme.html>.
- Rice, J. 1995.** *Mathematical Statistics and Data Analysis (Second ed.)*. s.l. : Duxbury Press, 1995.
- Rockafellar, R.T. 1970.** *Convex Analysis*. s.l. : Princeton University Press, 1970.
- Sarmiento, A., Durán, I., Aguilera, P., and Cruces, S. 2012.** A study of methods for initialization and permutation alignment for time-frequency domain blind source separation. *Independent Component Analysis for audio and biosignal applications*. s.l. : Intech, 2012.
- Schneider, R. 1993.** *Convex Bodies: The Brunn-Minkowski Theory*. s.l. : Encyclopedia of Mathematics and its Applications, 1993.
- Schreier, P. J. 2003.** *Improper complex random processes for signal processing and communications*. Colorado, USA : University of Colorado at Boulder Boulder, 2003.
- Schreier, P.J. and Scharf, L.L. 2010.** *Statistical Signal Processing of Complex-Valued Data*. s.l. : Cambridge University Press, 2010.
- Shahbazpanahi, S., Gershman, A.B., and Giannakis, G.B. 2006.** Semiblind multiuser MIMO channel estimation using capon and MUSIC techniques. *IEEE Transactions On Signal Processing*. 2006, Vol. 54, 9.
- Shalvi, O. and Weistein, E. 1994.** Universal methods for blind deconvolution

(Chapter 4). [book auth.] Simon Haykin. *Blind deconvolution*. s.l. : Prentice Hall, 1994.

Talwar, S., Viberg, M., and Paulraj, A. 1996. Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array - Part I: Algorithms. *IEEE Transactions on Signal Processing*. 1996, Vol. 44, 5.

Theis, F. J., Jung, A., Puntonet, C. G., and Lang, E. W. 2003. Linear Geometric ICA: fundamentals and algorithms. *Neural Computation*. 2003, Vol. 15.

Theis, F. J., Lang, E. W., and Puntonet, C. G. 2004. A geometric algorithm for overcomplete linear ICA. *Neurocomputing*. 2004, Vol. 56.

Therrien, W. 1992. *Discrete Random Signals and Statistical Signal Processing*. s.l. : Prentice-Hall, 1992.

Tome, A.M., Teixeira, A.R., Lang, E.W., Stadlthanner, K., Rocha, A.P., and Almeida, R. 2004. Blind source separation using time-delayed signals. *Proceedings IEEE International Joint Conference on Neural Networks*. 2004, Vol. 3.

Van der Veen, A. J and Paulraj, A. 1996. An analytical constant modulus algorithm. 1996, Vol. 44, 5.

Via, J., Palomar, D. P., Vielva, L., and Santamaria, I. 2011. Quaternion ICA From Second-Order Statistics. *IEEE Transactions on Signal Processing*. 2011, Vol. 59, 4.

Vrins, F., Verleysen, M., and Jutten, C. 2005. SWM: A class of convex contrasts for source separation. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. 2005.

Weil, W. 1975. On Mixed Volumes of Nonconvex Sets. *Proceedings of the American Mathematical Society*. 1975, Vol. 53, 1.

Widrow, B. and Stearns, S.D. 1985. *Adaptive Signal Processing*. s.l. : Prentice Hall, 1985.

Widrow, B. McCool, J., Ball, M. 1975. The complex LMS algorithm. *Proceedings of the IEEE*. 1975, Vol. 63, 4.

Xu, W. and Qiao, S. 2005. A Divide-and-Conquer Method for the Takagi Factorization. *Technical Report No. CAS 05-01-SQ*. Department of Computing and Software, McMaster University, 2005.

Zarzoso, V, Comon, P., and Phlypo, R. 2010. A contrast function for Independent Component Analysis without permutation ambiguity. *IEEE Transactions on Neural Networks*. 2010, Vol. 21, 5.

Zarzoso, V., Comon, P. 2006. *Alphabet-based deflation for blind source extraction in underdetermined mixtures*. Liverpool, UK: Proc. ICA Research Network International Workshop, 2006.

Zarzoso, V., Martín-Clemente, R., and Hornillo-Mellado, S. 2012. Independent component analysis based on first-order statistics. *Signal Processing*. 2012, Vol. 92.

INDEX OF FIGURES

- Figure 2.1** - This is a contour plot of the function defined in (2.65), and used in Example 2.5. Blue circumferences correspond to lower values of the cost function, where red circumferences correspond to high values..... 46
- Figure 2.2** - This is a contour plot of the cost function $f(z, z^*) = |z|^2$, using two types of the step size $\mu^{(k)}$ 48
- Figure 2.3** - Scatter plots for (left) circular, (center) proper but non-circular, and (right) improper, and thus non-circular, data. 49
- Figure 2.4** - This block diagram summarizes the discussion presented in Section 2.5.2. Also, we provide the relations between different representations and the equations where they are defined. 53
- Figure 3.1** - Block system of the sequence estimation problem. In the figure, $d(k)$ is the desired sequence that we want to estimate. $x(k)$ is the input sequence that we will use, by passing it by a linear system \mathbf{w} , resulting in the estimated sequence $y(k)$. The error between the estimate and the desired sequences is the estimation error $e(k)$, and it is used to design the coefficients of the filter. 61
- Figure 3.2** - This figure shows a two dimensional plane that represents the $\mathbf{x}(k)$ subspace. The solid line lying on the plane represents $y(k)$, the best possible approximation of the desired sequence $d(k)$ as a linear combination of $\mathbf{x}(k)$ with the weights \mathbf{w} . The difference between the desired and the output sequences is the error $e(k)$, which is orthogonal to the plane $\mathbf{x}(k)$ at the optimal solution..... 64

- Figure 3.3** - This is a 3D plot of the cost function $J_{LMS}(\mathbf{w})$, versus the plane conformed by the coefficients of the filter \mathbf{w} 66
- Figure 3.4** – Supergaussian and subgaussian distributions. A Gaussian distribution is plotted for comparison.76
- Figure 3.5** - Diagram of the linear mixing and extraction method for Independent Component Analysis. It is important to whiten the observations so the cost function is more tractable. Also, this improves the convergence and the robustness of the algorithms derived.77
- Figure 3.6** - This figure shows the comparison between the estimation square error of the Wiener (blue) and the Widely Linear Wiener (green) solutions to the speech separation problem, in the presence of additive noise.83
- Figure 3.7** - This figure shows the estimation of one source, using a linear extraction vector (lower right plot) and a widely linear one (lower left plot). The upper left plot is the spectrogram of the source. The upper right plot is the time information for some milliseconds, contained in one sub-band of the reference signal. The two lower plots show the amplitude versus time graphic of the local signal separation made by using a widely linear and a linear Wiener extraction vector.84
- Figure 4.1** - This plot shows a support set S_z defined by all the (red and black) points. The set of red points describes the convex hull S_z94
- Figure 4.2** - Graphical illustration of the Minkowski sum of two one dimensional sets and one planar set. For completeness, the boundary of the Minkowski sum of the first two one dimensional sets is also represented in dashed line in the final result.95
- Figure 4.3** - Graphical illustration of the Minkowski direct sum of a planar set and a one dimensional set. The sets lie in complementary linear subspaces, and the result lies in a space of higher dimension.96
- Figure 4.4** - This is a 3-dimensional representation of a convex set. Suppose that all the blue space is filled with points. The extreme points are shown in red. The number of extreme points on a convex set is relatively small, when compared with the total number of points on the set.98
- Figure 4.5** - This figure illustrates, for a 2×2 invertible mixture of two sources, the decompositions of the support of the observations as a Minkowski direct sum of two sets: $S_{a_1s_1}$ and $S_{a_2s_2}$. Since $m = 2$, the convex support of the observations is a set embedded in the four-dimensional real space. Hence, we have used a projection to represent it in only three dimensions. This projection was chosen so as to preserve invariant $S_{a_2s_2}$ while reducing the planar $S_{a_1s_1}$ set to a single dimension, as it is shown in the figure.100

Figure 4.6 - The definition of the convex perimeter (4.37) is illustrated here. The value ε is the distance between the outer and the inner set. The outer set is auxiliary and only exists to help in the definition of $\ell(S_y)$ 103

Figure 4.7 - This is a representation of the value of the cost function (4.53) versus the $m=2$ coefficients of the residual transfer vector. The left figure shows a 3D viewpoint, while the right figure shows a cenital viewpoint. For simplicity, all the values involved are real. Red zones correspond to high values, and blue zones correspond to low values. 107

Figure 4.8 - View of one of the polytopes $P_b(\rho, \theta)$ within the convex cone $C_b(\theta)$. The vertices of the polytope play a key role in the analysis of the solutions of the cost function. 108

Figure 4.9 - This figure shows conceptually the value of the cost function $\psi(\mathbf{b})$ on the polytope $P_b(\rho, \theta)$, and in part of the frontier of the convex cone $C_b(\theta)$ 109

Figure 4.10 - Block diagram of the mixing, whitening, and extraction systems. 113

Figure 4.11 - A set of samples from the output sequence y , with $T=14$. Observe how there are only $V=5$ vertices in the convex hull. 116

Figure 4.12 - This figure shows, on the left column, the output of the extraction algorithm $y(k)$. The red circles are the samples in the complex plane that belongs to the convex hull, and the edges $\Delta y_{(i)}$ are represented by a dashed line. On the right column, the absolute value of the global transfer vector $|\mathbf{g}|$ is shown. The upper row shows the state of the extraction at the 4-th iteration, and the lower row shows the state at the 40-th iteration..... 120

Figure 4.13 - First experiment. Example of the blind separation of sources by using a BCA extraction and deflation algorithm. The first column shows the original sources: a QPSK, a 16-QAM and a 32-QAM constellations. The second column shows the observations obtained after the mixing. Finally, the third column shows the extraction of the three original sources after a sequential extraction and deflation procedure..... 122

Figure 4.14 - Second experiment. Example of the blind separation of dependent sources by using a BCA extraction and deflation algorithm. The first column shows the three QPSK original sources. The second column shows the noisy observations obtained after the mixing. The third column shows the extraction of the original sources after a sequential extraction and deflation procedure..... 123

Figure 4.15 - Third experiment. This figure shows the result of executing the BSS methods based on BCA, ThinICA, JADE, and FastICA to a noisy mixture of dependent sources. (upper row) Comparison of the Amari Index (AI) versus the SNR. (lower row) Each one of the points corresponds to the absolute value of the coefficients of the global transfer vector $|\mathbf{G}|$, for each one of the compared methods, for a SNR=50 dB situation.

The permutation and the scale is corrected to show the successful extraction when the matrix is close to the identity.....124

Figure 4.16 – First experiment of the Latent Component Decomposition. This figure shows the result of a bounded component decomposition applied to a set of three observations, marked by blue dots. The bounded components are marked by green dots, while the residual unbounded component is marked by red dots.128

Figure 4.17 – Second experiment of the Latent Component Decomposition. This figure shows the result of a bounded component decomposition applied to a set of five observations, marked by blue dots. The bounded components are marked by green dots, while the residual unbounded component is marked by red dots.129

Figure 4.18 – Third experiment of the Latent Component Decomposition. The blue dots represent the three observations, while the green dots represent the bounded components decomposed sequentially by the algorithm presented in Table 4.2. At the top of the figure, the sources corresponding to the output of (4.104) are marked in red. The extraction vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are responsible for the conformation of the bounded components.....130

Figure 5.1 – Surface and contour plot of the cost function when considering a mixture $A = [1, 1; -1, 1]$ of two sources with respective perimeters: 1 and 0.8.....135

Figure 5.2 - This is a representation of the first-order local approximation used in (5.12) to derive the Newton-Raphson step size.....138

Figure 5.3 – This figure shows, in logarithmic scale, the comparison of the value of the distance to the convergence $\psi(\mathbf{u}^{(k)}) - \psi_{\min}$, when the Newton-Raphson step size (blue) and the step size that uses the second-order information (green) are used. We study two different cases: having 6 sources (dotted) and having 20 sources (solid). This plot was made assuming a noiseless situation, but the main effect of adding a random noise is the existence of a ground floor on the convergence curves.142

Figure 5.4 – This figure serves to illustrate the underlying geometrical interpretation of the BCA descent iteration, which has been addressed in Theorem 5.5. As we are in the two-dimensional plane, we presume a mixture of two sources ($n = 2$) with perimeters L_1 and L_2 . In the figure, the notation $P_{\mathbf{g}^{(k)}}^\perp$ has been used to abbreviate the orthogonal projector.....149

Figure 5.5 – Geometrical interpretation of the movement of the points (β_i, β_1) , for $i = 2, 3$, depending on the value of the step size. The blue arrows show the movement of the coefficients during two consecutive iterations. The black circles O, P, S, and B corresponds to (β_2, β_1) , while the black squares O', P', S', and B' corresponds to (β_3, β_1) , at any given iteration. The green dotted line shows the limit for the global

monotonous convergence (in the sense of the descent of $c_2^{(k)}$), which the moving points cannot exceed to satisfy the condition (5.81). The red dotted line shows the limit for the prevention of a leakage of the iteration to another basin of the cost function. 154

Figure 5.6 – First experiment. We show the remaining value of the normalized cost function, for four different step sizes. The NR step size does not converge in this example. The blue line corresponds to $\hat{\mu}_s^{(k)}$, the red line to the Newton-Raphson step size, and the green line to $\hat{\mu}_p^{(k)}$. The black dashed line corresponds to a fixed step size of value 0.01..... 165

Figure 5.7 – Second experiment. The blue arrows show the behavior of the g_1 and g_2 coefficients on the two dimensional plane, during several executions of the first iterations of the Bounded Component Analysis extraction algorithm. (left) A fixed step size $\mu_r = 0.01$ is used. (center) The Newton-Raphson step size is used. (right) The step size $\mu_s^{(k)}$ is used. The straight red lines correspond to the points of initialization where the movement is so strong to guarantee the convergence. The green dashed line corresponds to the direction of the vector \mathbf{L}_s , and divides the plane in two regions of convergence. 166

Figure 5.8 – Third experiment. This figure shows the behavior of the $\beta_i^{(k)}$ coefficients during the 100 iterations of the BCA extraction algorithm. (upper) Comparison between the $\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}$ coefficients when using the proposed step size (solid lines), and when using the Newton-Raphson step size (dotted lines). (lower left) The movement of the ten $\beta_i^{(k)}$ coefficients when using the proposed step size. (lower right) The movement of the ten $\beta_i^{(k)}$ coefficients when using the Newton-Raphson step size. 167

Figure 5.9 – Third experiment. This figure shows the convergence of the Newton-Raphson step size (right) and the proposed step size (left). 168

Figure 6.1 – This block diagram shows the signals and the systems presented in the training stage of a supervised signal extraction algorithm. 174

Figure 6.2 – This figure shows, in logarithmic scale, the comparison of the value of the cost function $\psi(\mathbf{b})$, when the Newton-Raphson (blue) and the step size that uses the second-order information (green) are used in a supervised extraction. We study two different cases: having 6 QPSK sources (dotted) and having 20 QPSK sources (solid). 182

Figure 6.3 - This figure shows the shape of the MSE (upper row) and minimum convex perimeter (lower row) cost functions. On the left column, the 3D representation of the surface ψ is shown. On the right column, a cenital view of the same function is shown. 184

Figure 6.4 - This figure shows the geometry of the 1-norm discussed in this section. The

surfaces of equally norm are rhombus. The axes are the coefficients g'_1 and g'_2 186

Figure 6.5 - This figure shows the geometry of the 2-norm discussed in this section. The surfaces of equally norm are circles. The axes are the coefficients g'_1 and g'_2 188

Figure 6.6 – Geometry of the discussion presented in Section 6.7.3. This time, the axes are the coefficients g'_1 , g'_2 , and g'_3 . (left) Geometry of the 1-norm in under-determined problems. (right) Geometry of the 2-norm in under-determined problems.189

Figure 6.7 – This figure shows the absolute value of the vector of extended coefficients, for three examples of over-determined systems. (upper) $n=3$ sources and $m=5$ observations. (middle) $n=4$ sources and $m=5$ observations. (lower) $n=8$ sources and $m=10$ observations. The first n coefficients correspond to the sources (1 desired signal and $n-1$ interference signals), and the last m coefficients correspond to noise. The exact and the minimum convex perimeter solutions are not normalized.197

Figure 6.8 – This figure represents the vector of extended coefficients for three examples of under-determined systems. (upper) $n=5$ sources and $m=3$ observations. (middle) $n=6$ sources and $m=5$ observations. (lower) $n=10$ sources and $m=8$ observations. The coefficients are sorted in these plots with the following structure: The first coefficient corresponds to the extracted source, the next $n-m$ coefficients correspond to the non-avoidable interference. The next $n-1$ coefficients correspond to the interference that can be cancelled, due to the available degrees of freedom, while the last m coefficients correspond to m noise signals. The exact and the minimum convex perimeter solutions are not normalized.198

Figure 6.9 - This figure represents the Performance Index (PI) versus the SNR, for the three examples of over-determined systems presented before. (left) $n=3$ sources and $m=5$ observations. (center) $n=4$ sources and $m=5$ observations. (right) $n=8$ sources and $m=10$ observations. The curve corresponding to the exact PZF solution is not shown because it is always zero, since it uses the perfect knowledge of the mixing matrix.200

Figure 6.10 - This figure represents the Performance Index (PI) versus the SNR, for the three examples of under-determined systems presented before. (left) $n=5$ sources and $m=3$ observations. (center) $n=6$ sources and $m=5$ observations. (right) $n=10$ sources and $m=8$ observations. The curve corresponding to the exact PZF solution is not shown because it is always zero, since it uses the perfect knowledge of the mixing matrix.201

Figure 6.11 – This figure shows the outputs of the extraction system in the complex plane, for three different under-determined mixtures. (left column) $n=5$ sources and $m=3$ observations. (center column) $n=6$ sources and $m=5$ observations. (right column) $n=10$ sources and $m=8$ observations. The first row (blue dots) corresponds to the MMSE solution, the second row (red dots) corresponds to the minimum convex perimeter solution, and the third row (green dots) corresponds to the exact solution.202

Figure 6.12 – This figure shows the block diagram of the complete system of Section 6.9.2. After the extraction, the output is passed through a decisor which has been trained with the same pilot symbols than the extraction vector. The result is a non-linear estimation of the original source..... 203

Figure 6.13 – The left figure shows the construction of the multimodal p.d.f. used in the MAP decisor for the symbol $1+j$. In the right figure, the output of a BCA extraction algorithm is shown, for an under-determined system with $n = 6$ and $m = 5$. The blue circles are the samples of $y(k)$ for the indices k that satisfy $s_1(k) = 1+j$ 204

Figure 6.14 – In this figure, the four surfaces built for the MAP decision are shown, for the three different solutions with PZF discussed during the whole chapter: the MMSE solution (upper), the minimum convex perimeter solution (middle) and the exact solution (lower). The under-determined mixture uses $n = 6$ sources and $m = 5$ observations, and the SNR = 15 dB. We took $T = 2000$ symbols for the training stage.205

Figure 6.15 – SER curves for the three solutions explored in this chapter, and two different under-determined experiments. (left) $n = 5$ sources and $m = 3$ observations. (right) $n = 7$ sources and $m = 5$ observations. The decision is based upon a MAP classifier. 206

Figure 6.16 - This plot shows the result of a classification of symbols by using a SVM. The green points are samples of the output $y(k)$ that corresponds to the input symbol $1+j$, while the magenta points corresponds to the input symbols $1-j, -1-j$, and $-1+j$. The upper row shows the output of the MMSE criterion, while the lower row shows the output of the minimum convex perimeter criterion. The right column zooms into a conflictive region on the complex plane. The decision bounds given by the SVM are shown as black lines rounding some regions..... 207

Figure 6.17 – SER curves for the three solutions explored in this chapter, and two different under-determined experiments. (left) $n = 5$ sources and $m = 3$ observations. (right) $n = 6$ sources and $m = 5$ observations. The decisor is based upon a SVM classifier. 209

INDEX OF TERMS

<i>absolute value</i>	30	<i>complex derivative</i>	37
<i>Additive White Gaussian Noise</i>	49	<i>complex differentiable</i>	35
<i>advantage of the widely linear MSE</i> . 82		<i>complex functions</i>	31
<i>analytic</i>	34	<i>Complex Least Mean Squares</i>	65
<i>augmented complex</i>	50	<i>complex linear transformation</i>	51
<i>augmented covariance matrix</i>	54	<i>complex variables</i>	24
<i>augmented gradient</i>	136	<i>Compressed Sensing</i>	178
<i>augmented Hessian</i>	136	<i>conjugate</i>	30
<i>augmented matrix</i>	52	<i>conjugate covariance matrix</i>	54
<i>augmented mean vector</i>	53	<i>conjugate gradient</i>	42
<i>auxiliary affine function</i>	178	<i>contrast function</i>	24
<i>blind</i>	24, 70	<i>convergence</i>	25
<i>Blind Source Extraction</i>	71	<i>convergence in the mean</i>	67
<i>Blind Source Separation</i>	70	<i>convergence in the mean square</i>	70
<i>Bounded Component Analysis</i>	90	<i>convex cone</i>	108
<i>Central Limit Theorem</i>	74	<i>convex hull</i>	95
<i>circular</i>	49	<i>convex perimeter</i>	102
<i>circularity coefficients</i>	85	<i>convex support</i>	95
<i>circularity spectrum</i>	86	<i>cost function</i>	24
<i>classifier</i>	203	<i>cost functions</i>	31
<i>closed-form solution</i>	64	<i>covariance</i>	53
<i>complementary covariance matrix</i>	54	<i>criterion of minimum normalized</i>	
<i>complex analytic</i>	35	<i>convex perimeter</i>	106

- cross-correlation* 72
cross-covariance..... 53
decisor.....49, 202
decorrelation 72
deflation..... 71
desired sequence 60
determined..... 91
differences..... 115
directly indecomposable..... 96
domain..... 31
Eigenvalue Decomposition..... 69
eigenvalues 69
estimation error 61
Euclidean distance..... 32
exact PZF solution 191
extended vector of coefficients..... 176
extended vector of sources 176
extraction vector..... 76
extreme points43, 98
global monotone convergence..... 133
global transfer vector 92
gradient 42
gradient descent algorithm..... 25
Hermitian matrix 41
holomorphic function..... 31
Hotelling transform..... 72
identity matrix 41
imaginary..... 30
improper 54
inner products 41
input sequence..... 60
isobars..... 45
iteration index 134
Joint Approximate Diagonalization of Eigenmatrices (JADE) 71
joint probability density function..... 74
jointly Gaussian.....74
Karhunen-Loève transform.....72
kurtosis.....75
Lagrange multipliers..... 192
Laplacian.....75
latent components 126
linear filter.....60
linear mixing model71
linear-conjugate-linear.....52
loss functions31
maximized.....44
Maximum A Posteriori (MAP).....203
Maximum Likelihood (ML)203
mean.....52
mean square61
Mean Square Error61
minimized44
Minimum Mean Square Error (MMSE)60
Minkowski algebra.....95
Minkowski direct sum96
Minkowski scalar product.....95
mixed volume..... 103
modes of convergence69
Moore-Penrose pseudo-inverse93
mutual independence74
non-stationary.....47
norm.....41
normalized convex perimeter90, 117
Normalized Least Mean Squares66
objective functions.....31
observations71
orthogonal matrix41
orthogonal projector.....148
orthogonality condition.....64
output sequence60

<i>output-sensitive</i>	121	<i>second order circularity</i>	49
<i>overdetermined</i>	91	<i>self-adjoint matrix</i>	41
<i>over-determined</i>	24	<i>semi-blind</i>	173
<i>Partial Zero Forcing (PZF)</i>	190	<i>semi-supervised</i>	174
<i>Performance Index (PI)</i>	199	<i>separation matrix</i>	71
<i>pilot symbols</i>	175	<i>Signal to Noise Ratio</i>	121
<i>polytope</i>	108	<i>Signal to Noise Ratio (SNR)</i>	191
<i>positive semidefinite</i>	68	<i>Singular Value Decomposition</i>	86
<i>power</i>	54	<i>sources</i>	70
<i>Principal Component Analysis</i>	72	<i>sphere decoding</i>	172
<i>principal components</i>	74	<i>stationary points</i>	43
<i>probability density function (p.d.f.)</i> ..	49	<i>statistical range</i>	111
<i>proper</i>	48	<i>steepest descent</i>	47
<i>pseudo-covariance</i>	54	<i>step size</i>	47
<i>quadratic form</i>	63	<i>stochastic gradient descent algorithm</i>	65
<i>Quadrature Amplitude Modulation (QAM)</i>	123	<i>strong uncorrelatedness</i>	56
<i>Quadrature Phase-Shift Keying</i>	49	<i>Strong Uncorrelating Transform (SUT)</i>	86
<i>quasiconcave</i>	108	<i>strongly uncorrelated</i>	86
<i>Radial Basis Function kernel</i>	208	<i>subgaussians</i>	75
<i>random processes</i>	48	<i>supergaussians</i>	75
<i>random variable</i>	48	<i>supervised</i>	24, 60, 173
<i>real</i>	30	<i>support</i>	94
<i>real composite</i>	50	<i>Support Vector Machine (SVM),</i> ...	208
<i>real composite linear</i>	51	<i>Symbol Error Rate (SER)</i>	206
<i>real conjugate derioative</i>	36	<i>symmetric matrix</i>	41
<i>real derivative</i>	36	<i>Takagi's factorization</i>	86
<i>real-to-complex</i>	51	<i>trace</i>	70
<i>Recursive Least Squares</i>	66	<i>train</i>	60
<i>regular</i>	35	<i>training error</i>).....	174
<i>regular mixing matrix</i>	112	<i>underdetermined</i>	91
<i>relation matrix</i>	54	<i>under-determined</i>	24
<i>remainder</i>	128	<i>under-determined</i>	172
<i>residual mixture vector</i>	92	<i>uniform</i>	75
<i>residual transfer matrix</i>	93	<i>unitary matrix</i>	41
<i>saddle point</i>	44	<i>unitary mixing matrix</i>	112

<i>unitary transformation</i>	69	<i>widely linear filter</i>	79
<i>whitened observations</i>	76	<i>Widely Linear Mean Square Error</i> ...	79
<i>whitened sequences</i>	72	<i>widely linear transformations</i>	51
<i>whitening</i>	72	<i>Wirtinger calculus</i>	37
<i>Wide Sense Stationary</i>	61	<i>Zero Forcing (ZF)</i>	172
<i>widely linear</i>	52		