SAPIENZA
UNIVERSITÀ DI ROMA

FACOLTÀ DI INGEGNERIA

DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE E DELLA COMUNICAZIONE
XXI CICLO

Tesi di DOTTORATO

Flexible Methods for Blind Separation of Complex Signals

Tutor
Prof. Aurelio Uncini

Candidato
Ing. Michele Scarpiniti

Anno Accademico 2007/2008

Flexible Methods for Blind Separation of Complex Signals

To Corinna

Anything that happens, happens.
Anything that, in happening, causes something else to happen, causes something else to happen.
Anything that, in happening, causes itself to happen again, happens again.
It doesn't necessarily do it in chronological order, though.
D. Adams

Everything should be made as simple as possible,
but not simpler.
A. Einstein

Abstract

ONE *of the main matter in Blind Source Separation (BSS) performed with a neural network approach is the choice of the nonlinear activation function (AF). In fact if the shape of the activation function is chosen as the cumulative density function (c.d.f.) of the original source the problem is solved. For this scope in this thesis a flexible approach is introduced and the shape of the activation functions is changed during the learning process using the so-called "spline functions".*

The problem is complicated in the case of separation of complex sources where there is the problem of the dichotomy between analyticity and boundedness of the complex activation functions. The problem is solved introducing the "splitting function" model as activation function. The "splitting function" is a couple of "spline function" which wind off the real and the imaginary part of the complex activation function, each of one depending from the real and imaginary variable.

A more realistic model is the "generalized splitting function", which is formed by a couple of two bi-dimensional functions (surfaces), one for the real and one for the imaginary part of the complex function, each depending by both the real and imaginary part of the complex variable.

Unfortunately the linear environment is unrealistic in many practical applications. In this way there is the need of extending BSS problem in the nonlinear environment: in this case both the activation function than the nonlinear distorting function are realized by the "splitting function" made of "spline function".

The complex and instantaneous separation in linear and nonlinear environment allow us to perform a complex-valued extension of the well-known INFOMAX algorithm in several practical situations, such as convolutive mixtures, fMRI signal analysis and bandpass signal transmission.

In addition advanced characteristics on the proposed approach are introduced and deeply described. First of all it is shows as splines are universal nonlinear functions for BSS problem: they are able to perform separation in anyway. Then it is

analyzed as the "splitting solution" allows the algorithm to obtain a phase recovery: usually there is a phase ambiguity. Finally a Cramér-Rao lower bound for ICA is discussed.

Several experimental results, tested by different objective indexes, show the effectiveness of the proposed approaches.

Contents

List of Figures

List of Tables

1

Introduction

—Since the real talent is having the answers
when the questions do not still exist.
A. Baricco

In the last years Blind Source Separation (BSS) realized through Independent Component Analysis (ICA) have raised great interest in the signal processing community [44, 77, 159]. In this context the neural network approach [76] (usually based on a single layer perceptron (SLP) or a multilayer perceptron (MLP)) seems to be one of the preferred methodologies [15, 94]; this interest is justified by the large number of different approaches and applications. As a matter of fact, in several fields, from multimedia to telecommunication and to biomedicine, ICA is currently employed to effectively recover the original sources from their mixtures or to remove interfering signals from the signal of interest. Initial studies on ICA aimed at solving the well-known cocktail party problem, in a instantaneous or slightly reverberant environment. Pioneering works in ICA appeared at the beginning of the 90's, when Jutten and Herault [94] presented their "neurometric architecture" and Comon [50] published his often referenced work.

Recently the problem of source separation has been extended to the complex domain [35, 21, 68], due to the need of frequency domain signal processing which is quite common in telecommunication [19] and biomedical applications [29, 30]. One of the most critical issues in ICA is the matching between the probability density function (or pdf) of sources (usually unknown) and the algorithm's parameters [218]. In this way one of the most important issues in designing complex neural networks consists in the definition of the complex activation function [20, 48, 101]. In order to improve the pdf matching for the learning algorithm, the so called Flexible ICA was recently introduced in [42, 67, 175, 208, 214]. Flexible ICA is the approach

in which the activation function (AF) of the neural network is adaptively modified during the learning. This approach provides faster and more accurate learning by estimating the parameters related to the pdf of signals. In literature it is possible to find several methods based on polynomials [5] and on parametric function approaches [146, 176].

Moreover the main properties that the complex activation function should satisfy [103, 214] are that it should be non linear and bounded and its partial derivatives should exist and be bounded. Unfortunately the analytic and boundedness characteristics are in contrast with the Liouville theorem [48, 101]. In other words, according to this theorem, an activation function should be bounded almost everywhere in the complex domain [48, 112, 69, 99, 101, 102, 2].

In this context, spline-based nonlinear functions seem to be particularly appealing as activation functions. In fact splines can model a very large number of nonlinear functions and can be easily adapted by suitably varying their control points, with low computational burden.

Unfortunately linear instantaneous mixing models are too unrealistic and unsatisfactory in many applications. Recent studies on ICA in the real domain showed that source separation can be effectively performed also in the case of convolutive nonlinear mixing environments [95, 208]. In the case of the complex domain only linear instantaneous mixtures have been considered so far [21, 201, 30, 2].

A more realistic mixing system inevitably introduces a nonlinear distortion in the signals. In this way the possibility of taking into account these distortions can give better results in signal separation. The problem is that in the nonlinear case the uniqueness of the solution is not guaranteed.

The solution becomes easier in a particular case, called Post Nonlinear (PNL) mixture, well-known in literature in the case of the real domain [183, 186]. In this context the solution is unique too. The work here exploited extends the linear and PNL mixture to the complex domain (complex-PNL). This extension requires proper modelling of the nonlinear distorting functions and of the activation functions of a feed-forward network. In this work this modelling has been performed by use of the splitting functions described in [200]. Another important issue is the definition of the theoretical conditions that grant the uniqueness of the solution.

1.1 The philosophy of BSS algorithm

ICA algorithms can be divided in two main and general classes: those based on *High Order Statistics* (*HOS*) which measure a "distance" from the gaussianity and those based on the *Information Theory* .

The HOS methods are based on a corollary of the Darmoi-Skitovitch theorem [57, 173] which asserts that at most one source signal can have

gaussian distribution: in this sense non-gaussianity has a key role in ICA. Effectiveness measures of non-gaussianity are the high order cumulants [33, 34, 37]. Usually the non-gaussianity is measured by the absolute value of the *kurtosis* for its simplicity in estimation (the square of kurtosis can also be used). These are zero for a gaussian variable and greater than zero for most non-gaussian variables. The main problem is that kurtosis can be very sensitive to outliers. Its value may depend on only a few observations in the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of nongaussianity.

A second very important measure of non-gaussianity is given by *negentropy* (see Appendix A.4), based on the information-theoretic quantity of (differential) entropy. The negentropy is always non-negative, and it is zero if and only for a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations: it is the optimal estimator of nongaussianity. The problem in using negentropy is that it is computationally very difficult. Estimating negentropy using the definition would require an estimate of the pdf. Therefore, some approximations have to be used.

On the other side a different approach is based on application of entropic contrasts, such as Joint Entropy (see Appendix A.1) and Mutual Information (see Appendix A.3) [15, 88, 90, 218, 219] which provides a simple and powerful approach to ICA.

The other side of the coin is the *maximum likelihood* (*ML*) approach [38, 33] which leads to the same results of the information theory approach [111].

1.2 History of ICA and BSS

The technique of ICA, although not yet the name, was introduced in the early 1980s by J. Hérault, C. Jutten, and B. Ans in [8].
The problem first came up in 1982 in a neurophyiological setting. A good historical introduction to ICA can be found in [96].

All through the 1980s, ICA was mostly known among French researchers, with limited influence internationally. The few ICA presentations in international neural network conferences in the mid-1980s were largely buried under the deluge of interest in back-propagation, Hopfield networks, and Kohonen's Self-Organizing Map (SOM), which were actively propagated in those times. Another related field was higher-order spectral analysis, on which the first international workshop was organized in 1989. In this workshop, early papers on ICA by J.-F. Cardoso [32] and P. Comon [49] were given. Cardoso used algebraic methods, especially higher-order cumulant tensors, which eventually led to the JADE algorithm [36]. The use of fourth-order cumulants has been earlier proposed by J.-L. Lacoume [106]. In signal processing literature, classic early papers by the French group are

[50, 52, 94, 179].

In signal processing, there had been earlier approaches in the related problem of blind signal deconvolution [59, 170].

The work of the scientists in the 1980's was extended by, among others, A. Cichocki and R. Unbehauen, who were the first to propose one of the presently most popular ICA algorithms [45, 46, 47]. Some other papers on ICA and signal separation from early 1990s are [24, 133]. However, until the mid-1990s, ICA remained a rather small and narrow research effort. Several algorithms were proposed that worked,usually in somewhat restricted problems, but it was not until later that the rigorous connections of these to statistical optimization criteria were exposed.

ICA attained wider attention and growing interest after A.J. Bell and T.J. Sejnowski published their approach based on the infomax principle [15, 14] in the mid-90's. This algorithm was further refined by S.-I. Amari and his co-workers using the natural gradient [5], and its fundamental connections to maximum likelihood estimation were established. A couple of years later, A. Hyvärinen, J. Karhunen and E. Oja presented the fixed-point or FastICA algorithm, [87, 86, 89], which has contributed to the application of ICA to large-scale problems due to its computational efficiency.

Since the mid-1990s, there has been a growing wave of papers, workshops, and special sessions devoted to ICA. The first international workshop on ICA was held in Aussois, France, in January 1999, and the second workshop followed in June 2000 in Helsinki, Finland. Both gathered more than 100 researchers working on ICA and blind signal separation, and contributed to the transformation of ICA to an established and mature field of research.

1.3 Organization

This dissertation aims to introduce a flexible solution to the *Blind Source Separation* problem and is organized as follows:

Chapter 2 introduces the Blind Source Separation (BSS) problem and some basic introductory concepts on ICA. Moreover the chapter explains the state of the art of the solution.

Chapter 3 explains the motivation and the usefulness in some practical situations of a complex representation of the data. Some scenarios of complex environment are given.

Chapter 4 extends the Blind Source Separation (BSS) problem and some basic introductory concepts on ICA to the complex domain. Moreover the chapter explains the state of the art of the complex BSS problem.

Chapter 5 introduces the problem of the complex activation function and

the properties which have to be satisfied. Different solutions of implementation of a such function are shown.

Chapter 6 introduces a flexible realization of the activation function realized through the so-called cubic spline functions. In this way the shape of the activation function can be adaptively changed during the learning process.

Chapter 7 introduces the de-mixing algorithms in the complex domain. Several metrics are adopted in order to reach a good convergence behavior. Moreover the different approaches ME and MMI are shown.

Chapter 8 explains some interesting results correlated to the previous algorithms. It is shown that splines are universal functions for the blind source separation problem and that the splitting solution allows the algorithm to recover the original phase. Moreover it shows a Cramér-Rao lower bound for ICA solution.

Chapter 9 shows several experimental results in different scenarios and demonstrates the effectiveness of the proposed approaches, both in real and in nonlinear environment.

Chapter 10 concludes the work and introduces some future other works and perspectives.

Appendix A introduces some elements of the information theory, which have been used in this thesis. In particular it defines the entropy of a random variable, the joint and conditional entropy of several random variables, the Kullback-Leibler divergence, the mutual information and the negentropy.

Appendix B introduces some elements of the complex variables and functions, which have been used in this thesis. In particular it is defined some useful statistical properties of complex random variables (r.v.s).

Appendix C presents a complete mathematical derivation of the learning rules for both ME and MMI approaches, and the demonstration of Theorem 17 in the thesis.

Appendix D aims to introduce some quite obscure mathematical concepts, which are sometimes not much known from usual readers, such as th Kronecker product.

2 Background

—A common mistake people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools.
D. Adams

LET us consider M observed signals $x_1[n], \ldots, x_M[n]$ at time n, which are assumed to be the mixtures of N independent source signals $s_1[n], \ldots, s_N[n]$ at time n. The vectors $\mathbf{s}[n] = [s_1[n], \ldots, s_N[n]]^T$ and $\mathbf{x}[n] = [x_1[n], \ldots, x_M[n]]^T$ are called the *source vector* and the *observation vector* respectively, T denotes the transpose operator. The observation vector is obtained by an unknown mixing system $\mathcal{F}\{\bullet\}$: $\mathbf{x}[n] = \mathcal{F}\{\mathbf{s}[n]\}$. In the general form, $\mathcal{F}\{\bullet\}$ may be nonlinear or may be convolutive. The goal of *Blind Signal Separation (BSS)* realized by conventional *ICA* is to construct a separating system $\mathcal{G}\{\bullet\}$ in order to obtain from the *output vector* $\mathbf{u}[n] = G\{\mathbf{x}[n]\}$ the estimate of the original source vector (see Figure 2.2) [90, 88, 159]. A classical example of sourse separation is the so-called *"cocktail party problem"* in which several people are speaking all together (see Figure 2.1).

In BSS only one a priori assumption is requested: sources must be statistically independent. However, the *probability density functions (pdf)* of the sources are usually unknown. The separating system $\mathcal{G}\{\bullet\}$ is constructed in such a way to obtain independent components, since the unique information about the sources is their statistical independence. This fact justifies the use of the *Independent Component Analysis (ICA)* approach in BSS.

The issue is to understand if the independence of the components of $\mathbf{u}$ implies necessarily the separation of the sources $\mathbf{s}$. Usually the approach to the problem is completely blind and no other knowledge is available on the mixing environment, it is possible only to make some hypotheses and provide the solution for this particular problem. In other words, considering

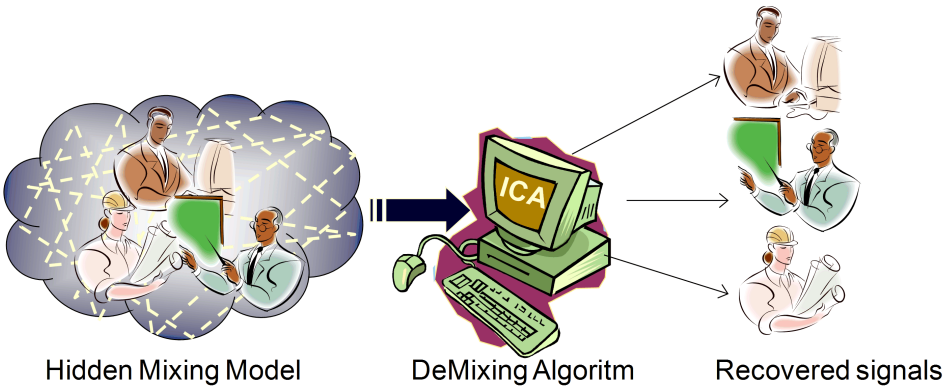

Fig. 2.1: The cocktail party problem

the space of all possible mixing environments, it is possible to grant that output independence produces the separation of signals only making some particular a priori assumption on the mixing environment.

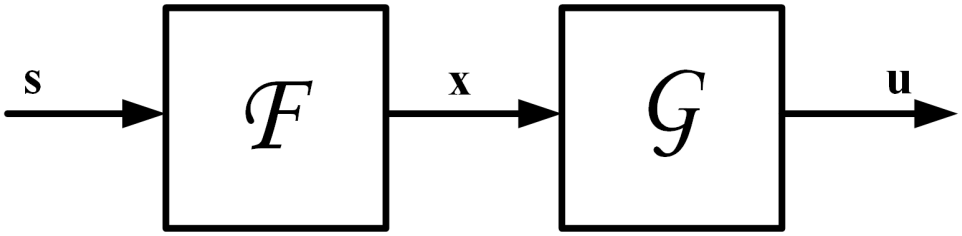

Fig. 2.2: Model of the mixing/de-mixing system

2.1 The case of linear mixing model

First of all the linear and instantaneous mixing model is introduced.

In linear environment both the mixing and the de-mixing models are linear. Under this condition the independence of the output insures the separation of the sources. In other words, linear instantaneous mixtures are separable. The separability and identifiability of the linear mixing model is presented in [50, 64, 190] as an application of the Darmois-Skitovich's theorem. A consequence of this result is that the vector s must have at most one Gaussian component.

By a linear instantaneous mixture we mean a mixture of the form

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] \tag{2.1}$$

where $\mathbf{A}$ is called the *mixing matrix*. Then a *separating* or *de-mixing matrix* $\mathbf{W}$ must be estimated to generate independent component outputs

$$\mathbf{u}[n] = \mathbf{W}\mathbf{x}[n] \tag{2.2}$$

where $\mathbf{u}[n]$ is an estimate of the source vector $\mathbf{s}[n]$ and its components are as independent as possible.

For simplicity we assume that the unknown mixing matrix is square ($M = N$). The linear mixing/de-mixing model is shown in Figure 2.3.

Although the solution is unique, it suffers for two kinds of ambiguities [90, 88]: it is not possible to determine the variances (energies) of the independent components, so we have a *scaling ambiguity* (in the complex case we have a rotation ambiguity due to the phase); it is not possible to determine the order of the independent components, so we have a *permutation ambiguity*. Formally these two ambiguities can be represented as a permutation $\mathbf{P}$ and a scaling $\mathbf{\Lambda}$ (diagonal) matrix:

$$\mathbf{WA} = \mathbf{P\Lambda} \tag{2.3}$$

Formally it is useful to give the following

Definition 1 *An invertible matrix $\mathbf{\Lambda}$ is said to be a scaling matrix, if it is diagonal. An invertible matrix $\mathbf{P}$ is said to be a permutation matrix, if it has exactly only one entry 1 in each row and each column and 0's elsewhere.*

2.2 The case of nonlinear mixing model

Since the linear mixing model is too poor and unrealistic in many applications, the complexity of the mixing model has been improved considering non linear models.

If the mixing-separating system is nonlinear and no other assumption is given for the mixing operator $\mathcal{F}\{\bullet\}$, a generic de-mixing model $\mathcal{G}\{\bullet\}$ does not assure the existence and uniqueness of the solution, so the separation is not guaranteed. Hence, in general, non-linear mixing models with no particular a priori assumptions are affected by a strong non-uniqueness [61, 94, 193].

In order to better illustrate this aspect, see the following example.

Example 1.

Consider two independent random variables s_1 with uniform distribution in $[0, 2\pi)$ and s_2 with Rayleigh distribution, so that its pdf is $p_{s_2}(s_2) = \frac{s_2}{\sigma_2^2} e^{-s_2^2/2}$ with variance $\sigma_2^2 = 1$ [144].
Given the two nonlinear transformations $y_1 = s_2 \cos s_1$ and $y_2 = s_2 \sin s_1$, the random variables y_1 and y_2 are still independent but are Gaussian distributed, so they cannot be separated as a consequence of the *Darmois-Skitovich's theorem* [95, 57, 173, 195]. In fact the Jacobian $\mathbf{J}$ of this transformation is:

$$\det(\mathbf{J}) = \det \begin{pmatrix} \frac{\partial y_1}{\partial s_1} & \frac{\partial y_1}{\partial s_2} \\ \frac{\partial y_2}{\partial s_1} & \frac{\partial y_2}{\partial s_2} \end{pmatrix} = \det \begin{pmatrix} -s_2 \sin s_1 & \cos s_1 \\ s_2 \cos s_1 & \sin s_1 \end{pmatrix} = -s_2$$

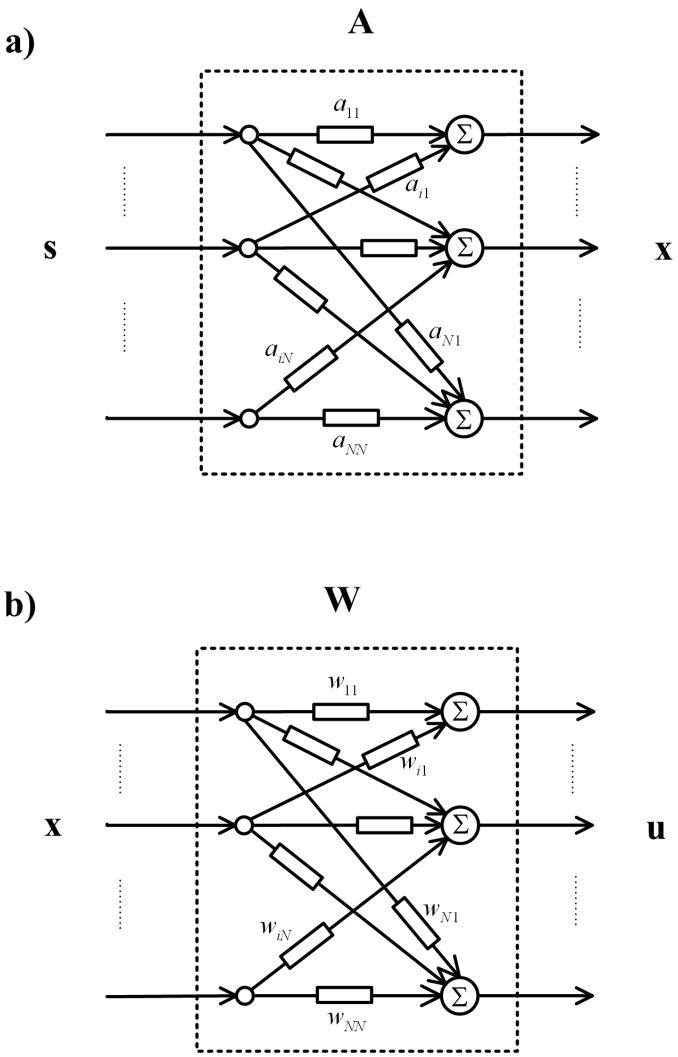

Fig. 2.3: Mixing (up) and de-mixing (down) models

so the joint pdf of $\mathbf{y} = [y_1, y_2]$ can be expressed as

$$p_{y_1,y_2}(y_1, y_2) = \frac{p_{s_1,s_2}(s_1,s_2)}{|\det(\mathbf{J})|} = \frac{1}{2\pi}\exp\left(-\frac{y_1^2+y_2^2}{2}\right) =$$
$$= \left(\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y_1^2}{2}\right)\right)\left(\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y_2^2}{2}\right)\right) \equiv p_{y_1}(y_1)\cdot p_{y_2}(y_2)$$

This simple example shows that in many cases the independence constraint is not strong enough to recover the original sources, unless additional assumptions about the transformation $\mathcal{F}(\bullet)$ or the mixing and de-mixing

model are taken.

In practice the main issue is to find the theoretical conditions in terms of sources, mixing environment and recovering architecture capable of guaranteeing the existence of the solution [193].

In [186] it has been presented an important result in nonlinear real valued environments: the solution to the BSS problem exists and is unique if we consider a particular mixing model called *Post Non-Linear (PNL) Mixtures*. This model consists in a cascade of a linear mixing stage and a set of nonlinear functions. Hence the mixing system $\mathcal{F}(\bullet)$ is (see left side of Figure 2.4):

$$\mathbf{x}[n] = F\{\mathbf{s}[n]\} = \mathbf{F}(\mathbf{A}\mathbf{s}[n]) = \mathbf{F}(\mathbf{v}[n]) \tag{2.4}$$

where $\mathbf{v}[n] = \mathbf{A}\mathbf{s}[n]$, the nonlinear function $\mathbf{F}(\mathbf{v}[n]) = [f_1(v_1[n]), \ldots, f_N(v_N[n])]^T$ is the model of the nonlinear distortion and $\mathbf{A}$ is an $N \times N$ matrix ($a_{ij} \in \mathbb{R}$).

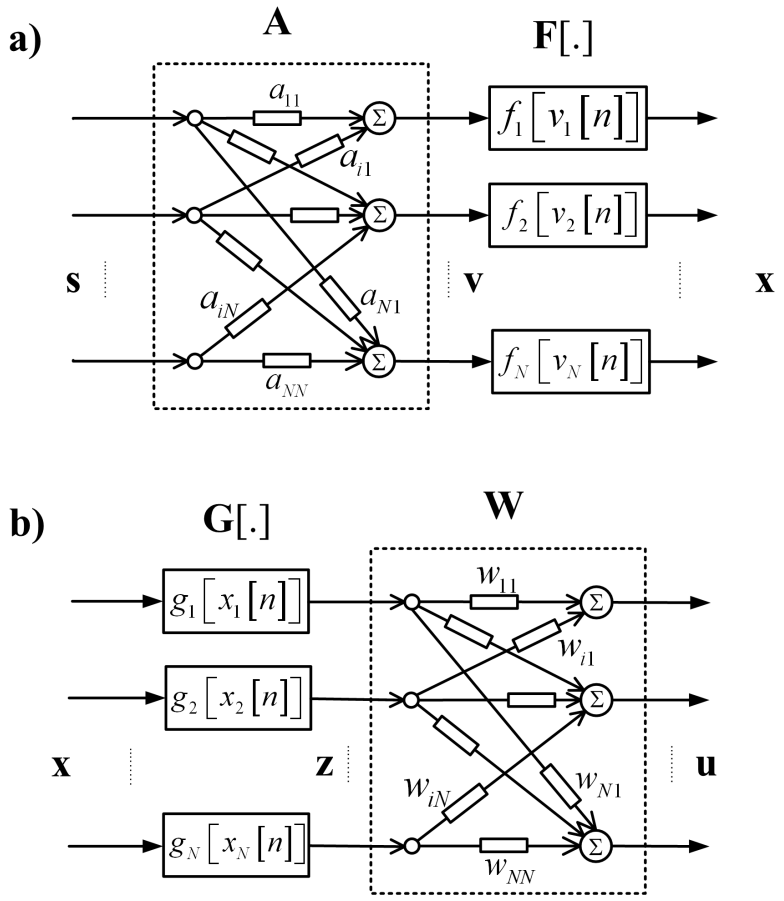

Fig. 2.4: The PNL model: mixing model (up) and de-mixing mirror model (down)

The de-mixing model $\mathcal{G}(\bullet)$ is constructed by the well-known *mirror model*:

the de-mixing system is the mirror image of the mixing one (see right side of Figure 2.4):

$$\mathbf{u}[n] = G\{\mathbf{x}[n]\} = \mathbf{W} \cdot \mathbf{G}(\mathbf{x}[n]) \tag{2.5}$$

where the nonlinear function $\mathbf{G}(\mathbf{x}[n]) = [g_1(x_1[n]), \ldots, g_N(x_N[n])]^T$ is the model of the nonlinear compensating functions and $\mathbf{W}$ is the de-mixing matrix.

Taleb and Jutten have demonstrated in [186] that if $\mathbf{A}$ and $\mathbf{W}$ are regular matrices, $f_i(v_i)$ $(i = 1, 2, \ldots, N)$ are differentiable invertible functions and $h_i = g_i \circ f_i$ satisfy the property that $h_i'(\vartheta) \neq 0$ ($\forall \vartheta \in \mathbb{R}$ and $\forall i = 1, 2, \ldots, N$) then the PNL mixtures are separable. However the proof in [186] contains an inaccuracy which was solved by Theis and Gruber in [193].

2.3 Basic concepts

At this time is very useful introducing some basic concepts and definition on the identifiability, separability and uniqueness of the model above described.

For $m, n \in \mathbb{N}$ let $Mat(m \times n; \mathbb{R})$ be the space of real $m \times n$ matrices, and

$$Gl(n) \equiv Gl(n; \mathbb{R}) = \{W \in Mat(n \times n) \mid \det(W) \neq 0\}$$

be the general linear group of $\mathbb{R}^n$.

Fundamental definitions on the matrix involved in the BSS problem can be done [188, 195].

Definition 2 *Let $\mathbf{A} \in Gl(n)$ be an invertible matrix. Then $\mathbf{A}$ is said to be mixing if $\mathbf{A}$ has at least two nonzero entries in each row.*

Definition 3 *We say two matrices $\mathbf{A}, \mathbf{W} \in \mathbf{Mat}(\mathbf{m} \times \mathbf{n})$ are **equivalent**, $\mathbf{W} \sim \mathbf{A}$, if $\mathbf{W}$ can be written as $\mathbf{W} = \mathbf{APA}$ with an invertible diagonal matrix (scaling matrix) $\mathbf{\Lambda} \in Gl(n)$ and an invertible matrix with unit vectors in each row (permutation matrix) $\mathbf{P} \in Gl(n)$. Note that permutation and scaling matrices commute, so $\mathbf{W} = \mathbf{APA} = \mathbf{A\Lambda P}$.*

And similarly

Definition 4 *$\mathbf{A}$ is said to be **scaling-equivalent** to $\mathbf{W}$, $\mathbf{A} \sim_{\mathbf{s}} \mathbf{W}$, if $\mathbf{W} = \mathbf{A\Lambda}$ holds, and $\mathbf{A}$ is **permutation-equivalent** to $\mathbf{W}$, $\mathbf{A} \sim_{\mathbf{p}} \mathbf{W}$, if $\mathbf{W} = \mathbf{AP}$. Therefore, if $\mathbf{A}$ is scaling- or permutation-equivalent to $\mathbf{W}$, it is equivalent to $\mathbf{W}$, but not vice-versa.*

In order to compare two $m \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ in $Mat(m \times n)$ with respect to equivalence as defined above, we calculate the generalized *crosstalking error* $E(\mathbf{A}, \mathbf{B})$ of $\mathbf{A}$ and $\mathbf{B}$ defined by

$$E(\mathbf{A}, \mathbf{B}) := \min_{\mathbf{M} \in \Pi(n)} \|\mathbf{A} - \mathbf{BM}\|,$$

where $\|\bullet\|$ is a fixed matrix norm and $\Pi(n)$ consists of all matrices with exactly one non-zero entry in each row and each column.

Lemma 1 *$E(\mathbf{A},\mathbf{B}) = 0$ if and only if $\mathbf{A}$ is equivalent to $\mathbf{B}$.*

Proof. Note that $\mathbf{\Pi}$ consists of all $n \times n$-matrices of the type $\mathbf{\Lambda P}$, where $\mathbf{\Lambda}$ is a non-degenerated diagonal matrix (scaling matrix) and $\mathbf{P}$ a permutation matrix. If $\mathbf{A}$ is equivalent to $\mathbf{B}$, then by definition there exists a $\mathbf{M} \in \mathbf{\Pi}$ such that $\mathbf{A} = \mathbf{BM}$, therefore $E(\mathbf{A},\mathbf{B}) = \mathbf{0}$. Vice versa, if $E(\mathbf{A},\mathbf{B}) = \mathbf{0}$, then there exists $\mathbf{M} \in \mathbf{\Pi}$ with $\|\mathbf{A} - \mathbf{BM}\| = 0$, i.e. $\mathbf{A} = \mathbf{BM}$. ■

In BSS problem a helpful index able to estimate the performance of separation will be introduced by eq. (9.2), which is rewritten here for simplicity:

$$S(Q) = \sum_{i=1}^{N} \left(\frac{\sum\limits_{k=1}^{N} |q_{ik}|^2}{\max\limits_{p} \left[|q_{ip}|^2 \right]} - 1 \right) + \sum_{k=1}^{N} \left(\frac{\sum\limits_{i=1}^{N} |q_{ik}|^2}{\max\limits_{p} \left[|q_{pk}|^2 \right]} - 1 \right). \tag{2.6}$$

where q_{ij} is the element of the matrix $\mathbf{Q} = \mathbf{WA}$ and is known as the **performance index**, often called **crosstalking error** [5] (see Section 9.1).

For the index (2.6) the following result holds

Lemma 2 *Let $\mathbf{Q} \in Gl(n)$. $S(Q) = 0$ if and only if $\mathbf{Q} \in \mathbf{\Pi}$, i.e. if $\mathbf{Q}$ is the product of a scaling and a permutation matrix.*

Proof. $\mathbf{\Pi}$ consists of matrices with exactly one nonzero element per column and per row. As $\mathbf{Q}$ is invertible, $\mathbf{Q}$ has at least one nonzero element per column and row, and $S(Q) = 0$ obviously if and only if $\mathbf{Q}$ is of that type, i.e. if $\mathbf{Q} \in \mathbf{\Pi}$. ■

With reference to the model (2.1) the couple $\mathbf{A}, \mathbf{s}$ is called a *representation* of r.vc. $\mathbf{x}$. It is also assumed that representations are *reduced* in the sense that columns in mixing matrices are not *pairwise* linearly dependent. No more is assumed a priori about the ranks and the number of columns.

It is easily shown [97] that if $\mathbf{x} = \mathbf{As} + \mathbf{a} = \mathbf{Br} + \mathbf{b}$ for some constant vectors $\mathbf{a}$ and $\mathbf{b}$, then linear manifolds generated by the columns of $\mathbf{A}$ and $\mathbf{B}$ coincide and the vector $\mathbf{a} - \mathbf{b}$ belongs to this common manifold. Therefore, $rank[\mathbf{A}] = rank[\mathbf{B}]$, and adding constants to the model (2.1) gives no additional generality since the constant could not be determined. Also models of the form $\mathbf{As} + \mathbf{Br}$, where $\mathbf{s}$ and $\mathbf{r}$ are independent and are comprised of independent variables, may be written as $(\mathbf{AB})(\mathbf{s^T r^T})^\mathbf{T}$. Thus the usual "noisy" ICA model $\mathbf{x} = \mathbf{As} + \mathbf{n}$, where $\mathbf{n}$ obeys multinormal distribution (i.e., linear transformation of independent normal variables), is just a special case of the general linear model (2.1).

The model of (2.1) is defined to be:

1. *identifiable*, or the mixing matrix is (essentially) unique, if in every reduced representations $(\mathbf{A}, \mathbf{s})$ and $(\mathbf{B}, \mathbf{r})$ of $\mathbf{x}$, every column of $\mathbf{A}$ is linearly dependent on some column of $\mathbf{B}$ and vice versa;

2. *unique* if the model is identifiable and further source r.vc.'s $\mathbf{s}$ and $\mathbf{r}$ have the same distribution for some permutation up to changes of location and scale;

3. *separable*, if for every matrix $\mathbf{W}$ such that $\mathbf{Wx}$ has m independent components, we have $\mathbf{\Lambda P s} = \mathbf{Wx}$ for some block diagonal matrix $\mathbf{\Lambda}$ with nonzero diagonals and permutation matrix $\mathbf{P}$. Moreover, such a matrix $\mathbf{W}$ has to always exist.

It follows from the reduction assumption that the number of columns, i.e., the number of sources or the *model order*, is the same in every representation of $\mathbf{x}$ in identifiable models. When there are more sources than mixtures, the problem of estimating the model order is, to our best knowledge, largely untackled.

If $\mathbf{W}$ is a separating matrix, then linear manifolds of $\mathbf{\Lambda P}$ and $\mathbf{W}$ must coincide, and therefore $p \geq rank[\mathbf{W}] = rank[\mathbf{\Lambda P}] = m$, i.e., there has to be at least as many mixtures as sources in a separable model. By only requiring that less than m sources are recovered, the separation in this sense becomes possible [31] for some models with $p < m$. Any separable model will be shown to be unique. However, the opposite is not true. In order to give more intuition to the definitions, consider the following examples.

Example 2.

If components of $\mathbf{s}$ are i.i.d. normally distributed, then also $\mathbf{\Lambda U s}$ has independent components (and is Gaussian) for any orthogonal matrix $\mathbf{U}$ and diagonal matrix $\mathbf{\Lambda}$. Therefore, any multinormal mixing is not identifiable.

Example 3.

As an example of a model which is identifiable but is not separable nor unique, consider independent nonnormal r.v.'s s_k, $k = 1, \ldots, 4$. Let n_1 and n_2 be standard normal and independent. Then also $n_1 + n_2$ and $n_1 - n_2$ are independent. Now

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 + n_1 \\ s_4 + n_2 \end{pmatrix} =$$

$$= \begin{pmatrix} s_1 + s_3 + s_4 + n_1 + n_2 \\ s_1 + s_3 - s_4 + n_1 - n_2 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 + n_1 + n_2 \\ s_2 + n_1 - n_2 \\ s_3 \\ s_4 \end{pmatrix}$$

which shows that the model can not be unique. However, it is identifiable. R.v.'s of the form $s + n$, where n is Gaussian, are said to have a *normal component*.

2.4 Identifiability, Separability, and Uniqueness of Linear ICA Models

Main of the result of this paragraph can be found in [64, 62, 189, 195] and traced back to [50].

Two theorems characterizing the distribution of r.vc.'s with linear structure of independent r.v.'s are presented in this section. These theorems were mainly derived in mathematical statistics community in the 1960s. For extensive treatment, additional theorems and proofs, see [97].

Theorem 1 *Let* $(\mathbf{A}, \mathbf{s})$ *and* $(\mathbf{B}, \mathbf{r})$ *be two representations of a* p*-dimensional r.vc.* $\mathbf{x}$*, where* $\mathbf{A}$ *and* $\mathbf{B}$ *are constant matrices of dimensions* $p \times m$ *and* $p \times n$ *respectively,* $p \geq 2$*, and* $\mathbf{s} = (s_1, \ldots, s_m)^T$ *and* $\mathbf{r} = (r_1, \ldots, r_n)^T$ *are r.vc.'s with independent components. Then the following properties hold.*

1. *If the* i*-th column of* $\mathbf{A}$ *is not linearly dependent on any column of* $\mathbf{B}$*, then* s_i *is normal.*

2. *If the* i*-th column of* $\mathbf{A}$ *is linearly dependent on the* j*-th column of* $\mathbf{B}$*, then the logarithms of the c.f.'s of* s_i *and* r_j *differ by a polynomial in a neighborhood of the origin.*

In order to introduce the second theorem, we first introduce the (column-wise) *Khatri-Rao product* $\odot$ on matrices defined as matrix column-wise *Kronecker product* $\otimes$ (for the definition of such operators, see Appendix D). If $\alpha_{\mathbf{1}}, \ldots, \alpha_{\mathbf{m}}$ and $\beta_{\mathbf{1}}, \ldots, \beta_{\mathbf{m}}$ are columns of $\mathbf{A}$ and $\mathbf{B}$, respectively, then $\mathbf{A} \odot \mathbf{B} = (\alpha_{\mathbf{1}} \otimes \beta_{\mathbf{1}} \cdots \alpha_{\mathbf{m}} \otimes \beta_{\mathbf{m}})$. The power $(\mathbf{A}\odot)^q\mathbf{A}$ is given naturally by $\mathbf{A} \odot \cdots \odot \mathbf{A}$ (includes q times $\odot$) [97].

Theorem 2 *Let a* p*-dimensional r.vc.* $\mathbf{x}$ *with nonvanishing c.f. have a representation* $(\mathbf{A}, \mathbf{s})$*, where* $\mathbf{A}$ *is a known* $p \times m$ *matrix and let* q *be the integer such that* $rank[(\mathbf{A}\odot)^q\mathbf{A}] = m > rank[(\mathbf{A}\odot)^{q-1}\mathbf{A}]$*. Then the c.f. of each r.v.* s_i *is determined up to a factor* $exp(\mathcal{P}_{i,q}(t))$*, where* $\mathcal{P}_{i,q}(t)$ *is a polynomial of degree at most* q*.*

2.4.1 Identifiability

In this section, the requirements for the identifiability of an ICA model are given in a form of a theorem. In an identifiable ICA model the coefficients of the mixing matrix may be determined from the mixtures alone up to permutation and scaling of columns. The fact that this is also possible in cases where we have more sources than mixtures is recognized by several authors in the ICA community.

Theorem 3 (Identifiability) *The model of* (2.1) *is identifiable among all representations* $(\mathbf{B}, \mathbf{r})$ *of* $\mathbf{x}$ *that*

1. *do not contain any Gaussian source;*
2. $\mathbf{B}$ *is of full column rank and at most one source r.v. is normal.*

It can be seen that the model in Example 4 is identifiable. The theorem is further illustrated by the following example.

Example 4.
Consider independent nonnormal r.v.'s s_1, s_2, and standard normal r.v.'s n_1 and n_2. Now

$$\mathbf{x} = \begin{pmatrix} s_1 + s_2 + 2n_1 \\ s_1 + 2n_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 + 2n_1 \\ 2n_2 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} s_1 + n_1 + n_2 \\ s_2 \\ n_1 - n_2 \end{pmatrix}$$

and the last column shows that the model is not identifiable. This shows why in general not a single normal r.v. is allowed for identifiability.

2.4.2 Separability

Separability considers the traditional linear ICA model and recovery of the sources. The following well-known theorem was proved in [31, 50] assuming r.v.'s with finite second order moments. The theorem given in this section relaxes the requirements imposed on the existence of the moments, and thereby extends the applicability of ICA to the systems where sources may have heavy-tailed distributions, [135].

Theorem 4 (Separability) *The model of* (2.1) *is separable if and only if the mixing matrix* $\mathbf{A}$ *is of full column rank and at most one source variable is normal.*

2.4.3 Uniqueness

Theorem 3 shows that if the number of sources is greater than the number of mixtures and the sources are non-Gaussian, it is still possible to identify the mixing matrix from the knowledge of $\mathbf{x}$ alone. However, it is not possible to recover the sources $\mathbf{s}$ (separability). The question arises if it is possible to determine also the distribution of $\mathbf{s}$ in such cases, i.e., are the models unique.

Uniqueness has also been implicitly used in the case when all sources are assumed to be discrete [51, 143]. However, theoretical justification has so far existed only for the finite alphabet discrete case [185], and there are no other known conditions that guarantee, for instance, the uniqueness of the likelihood function. The theorem given here extends the results of [185] to cases where the sources are not necessarily discrete r.v.'s. It is said that c.f. Φ has an exponential factor with a polynomial $\mathcal{P}$, if Φ can be written as $\Phi = \varphi exp(\mathcal{P})$ for some c.f. φ.

Theorem 5 (Uniqueness) *The model of* (2.1) *is unique if any of the following properties hold.*

1. *The model is separable.*
2. *All c.f.'s of source r.v.'s are analytic (or all c.f.'s are nonvanishing), and none of the c.f.'s has an exponential factor with a polynomial of degree at least 2.*
3. *All source r.v.'s are nonnormal with nonvanishing c.f.'s, and* $rank[\mathbf{A} \odot \mathbf{A}] = m$.
4. *All source r.v.'s have nonvanishing c.f.'s without exponential factors with a polynomial of degree* n, $1 < n \leq q$, *and* $rank[(\mathbf{A}\odot)^q\mathbf{A}] = m > rank[(\mathbf{A}\odot)^{q-1}\mathbf{A}]$.

R.v.'s with analytic c.f.'s [116] are exactly those for which the moment generating function exists (i.e., implies all moments exist). Analytic c.f.'s have only factors which are analytic c.f.'s [116], and by Theorem of Marcinkiewicz [116] c.f.'s of the form $exp(\mathcal{P}(u))$ are necessarily normal (or degenerate). Therefore, the analytic part of Case 2 in the above theorem could be reformulated as that the model is unique if all c.f.'s are analytical and none of the r.v.'s has a normal component (see Example 2).

Since the c.f. of a finite alphabet discrete r.v. does not contain an exponent factor with a polynomial of degree more than one, Case 2 covers the finite discrete distributions. This was proved in [185]. Additionally all r.v.'s with distribution function that is bounded from the left and right e.g., uniform, rectangular, and all truncated distributions, are covered by the property.

The number of source r.v.'s is unlimited in Case 2 of Theorem 5 for any number of mixtures. In the last two cases, the maximum number is limited. Since the number of independent rows in $\mathbf{A} \odot \mathbf{A}$ can be at most $p(p+1)/2$,

the maximum number m of sources for a given the number of mixtures p in Case 3 is $p(p+1)/2$, and the minimum value p of mixtures for given the number of sources m is obtained from $p(p-1)/2 < m \leq p(p+1)/2$. It is easily shown that these numbers are attainable, i.e., there exist matrices that fulfill the requirement. Case 4 extends uniqueness to even more source r.v.'s (with fewer mixtures) at the expense of fewer allowed distributions.

2.5 On the definition of linear ICA models

Given a random vector independent component analysis (ICA) tries to find its statistically independent components. This idea can be used also to solve the blind source separation (BSS) problem which is, given only the mixtures of some underlying independent source signals, to separate the mixed signals. It is necessary to introduce the definition of ICA rigorously [188]. In this and in the following paragraphs we report the main results from [188].

In independent component analysis, a random vector $X : \Omega \rightarrow \mathbb{R}^m$ called *mixed vector* is given, and the task is to find a transformation $f \circ X$ of X out of a given analysis model such that X is as independent as possible.

In order to specify this more precisely, we need to introduce terminology for the meaning of *degree of independence*. For this we introduce a *contrast function*. Suppose, we are given an analysis model $\tau \subset \{f : \mathbb{R}^m \rightarrow \mathbb{R}^n \mid f \; measurable\}$ and a real function κ called *contrast function* defined on random vectors, then the goal is to optimize $f \mapsto \kappa(f \circ X), f \in \tau$.

A contrast is a measure of how elements of an analysis model can transform a given random vector.

Definition 5 (Contrast) *A contrast of an analysis model τ is a function $\kappa : D_\kappa \rightarrow \mathbb{R}$, $D_\kappa \subset L^1(\Omega, \mathbb{R}^n)$, such that the following holds:*

1. *κ is a a probability theoretic notion, i.e. $\kappa(X) = \kappa(Y)$ if $X, Y \in D_\kappa$ with $P_X(X) = P_Y(Y)$ almost sure.*

2. *τ is closed with respect to D_κ that is if $g \circ X \in D_\kappa$ then $g' \circ X \in D_\kappa$ for all $X \in L^1(\Omega, \mathbb{R}^m)$ and $g, g' \in \tau$ ($\tau\tau^{-1}D_\kappa \in D_\kappa$).*

3. *$PX \in D_\kappa$ and $\kappa(PX) = \kappa(X)$ for any permutation $\mathbf{P}$ and $X \in D_\kappa$.*

4. *$\kappa(g \circ X) \geq \kappa(g' \circ X)$ for all $g' \in \tau$ if $g \circ X$ is independent.*

*The contrast is said to be **scaling-invariant**, if $LX \in D_\kappa$ and $\kappa(LX) = \kappa(X)$ for all invertible diagonal matrices $\mathbf{L}$ and $X \in D_\kappa$. We say that κ is **translation-invariant**, if $X + c \in D_\kappa$ and $\kappa(X + c) = \kappa(X)$ for all $c \in \mathbb{R}^n$ and $X \in D_\kappa$. Furthermore, κ is said to be **symmetry-invariant**, if $sym(X) \in D_\kappa$ and $\kappa(sym(X)) = \kappa(X)$ for all $X \in D_\kappa$.*

Transformations $g \in \tau$ that make X independent ($g \circ X$ independent) are maxima of κ_X. If κ is discriminating then those are the only maxima of κ_X:

Definition 6 *A contrast κ of τ is called discriminating if for all $X \in D_{(\tau,\kappa)}$ the following holds: $\kappa_X(g) \geq \kappa_X(g')$ for all $g' \in \tau$ then $g \circ X$ is independent.*

Independent Component Analysis deals with the question of how to transform random vectors to make them as independent as possible. Using the terminology of a contrast it is now possible to formulate what this should mean in mathematical terms.

Definition 7 (ICA model) *A pair (τ, κ) with an analysis model τ i.e. a subset $\tau \subset \{g : \mathbb{R}^m \to \mathbb{R}^n | g\ measurable\}$ and a contrast κ of τ is called an **independent component analysis model** (ICA model). We then denote the analysis model τ by **demixing model**. The ICA model is called **overcomplete** or **underdetermined** if $m < n$; if $m > n$ the model is called **undercomplete** or **overdetermined**; if $m = n$ we say the model is **symmetric**.*

Definition 8 (ICA) *Let $X : \Omega \to \mathbb{R}^m$ be a random vector and (τ, κ) an ICA model. An **independent component analysis (ICA)** of (X, τ, κ) is an element $g \in \tau$ such that $g \circ X \in D_\kappa$ and $\kappa(g \circ X) = \kappa(g' \circ X)$ for all $g' \in \tau$ with $g' \circ X \in D_\kappa$. Then $g \circ X$ is called an **ICA vector** of (X, τ, κ).*

Denote $ICA(X, \tau, \kappa)$ the set of all ICAs of (X, τ, κ).

Definition 9 (ICA algorithm) *Let (τ, κ) be an ICA model and $D \subset D_{(\tau,\kappa)} \subset L^1(\Omega, \mathbb{R}^m)$. A map $\iota : D \to \tau$ is called an **ICA algorithm** of (τ, κ) if $\iota(X)$ is an ICA of (X, τ, κ) for all $X \in D$.*

So an ICA algorithm says how to construct ICAs for a given set of random vectors X. Indeed a given algorithm can sometimes be extended to a larger definition set as the following lemma show.

Lemma 3 *Let $\iota : D \to \tau$ be an ICA algorithm with $D \subset L^1(\Omega, \mathbb{R}^m)$ and (τ, κ) a linear ICA model such that every $X \in D$ is centered. Define $D^* := \{X + c | X \in D|, c \in \mathbb{R}^m\}$. Then ι induces a map*

$$\begin{aligned} \iota^* : D^* &\to \tau \\ X &\mapsto \iota^*(X) := \iota(X - E(X)) \end{aligned}$$

If κ is translation-invariant then ι^* is an ICA algorithm of (τ^*, κ).

This lemma states that in linear ICA we only have to construct ICA algorithms for centered random vectors - as long as the contrast is translation invariant (which is the case for the mutual information for example).

Lemma 4 *Let $\iota : D \to \tau$ be an ICA algorithm with $D \subset L^1(\Omega, \mathbb{R}^m)$ and (τ, κ) a linear ICA model such that every $X \in D$ has a symmetric density. Define $D^* := \{X | sym(X) \in D\}$. Then ι induces a map*

$$\begin{aligned} \iota^* : D^* &\to \tau \\ X &\mapsto \iota^*(X) := \iota(sym(X)) \end{aligned}$$

If κ is symmetry-invariant then ι^* is an ICA algorithm of (τ^*, κ).

This means that in the case of a symmetry-invariant contrast we can furthermore restrict ourselves to symmetric random vectors when constructing ICA algorithms in linear ICA.

Definition 10 *Let $X : \Omega \to \mathbb{R}^m$ be a random vector and (τ, κ) an ICA model. Then X is said to be separable in (τ, κ) if there exists an ICA g of (X, τ, κ) with $g \circ X$ independent.*

Note that ICA generalizes PCA in the following sense:

Lemma 5 *An ICA of a separable X in (τ, κ), where κ is discriminating, is also a PCA of (X, τ).*

2.6 On the definition of linear BSS

In blind source separation, a random vector $X : \Omega \to \mathbb{R}^m$ called *mixed vector* is given; it comes from an independent random vector $S : \Omega \to \mathbb{R}^n$ which will be called *source vector*, by mixing with a *mixing function* $\mu : \mathbb{R}^n \to \mathbb{R}^m$, ie. $X = \mu \circ S$. Only the mixed vector is known, and the task is to recover μ and then S, so the task is to find an ICA of X.

In the symmetric case $(m = n)$, μ is usually assumed to be invertible, so reconstruction of μ directly gives S via $S = \mu^{-1} \circ X$. This means that if we assume that the inverse of the mixing function lies already in the transformation space $(\mu^{-1} \in \tau)$, then we know that the global maximum of the canonical contrast function has value 0, so indeed a global maximum will give us an independent random vector. Of course we cannot hope that μ^{-1} will be found because uniqueness in this general setting cannot be achieved. This will usually impose a restrictions on the analysis model.

Definition 11 (BSS model) *A **blind source separation model (BSS model)** is a triple (f, τ, κ) with $f : \mathbb{R}^n \to \mathbb{R}^m$ and (τ, κ) an ICA model such that there exists an independent random vector $S \in L^1(\Omega, \mathbb{R}^n)$ with $f \circ S \in D_{\tau,\kappa}$. f is called the **mixing mapping**.*

Definition 12 (BSS) *Let (f, τ, κ) be a BSS model and $S \in L^1(\Omega, \mathbb{R}^n)$ independent, called **source vector**. A **blind source separation (BSS)** of (S, f, τ, κ) is an ICA of $(f \circ S, \tau, \kappa)$.*

So, in BSS $f \circ X$ as well as τ and κ are known and the goal is to find estimates of f and S.

Denote $BSS(S, f, \tau, \kappa)$ the set of all BSSs (S, f, τ, κ), so by definition

$$BSS(S, f, \tau, \kappa) = ICA(f \circ S, \tau, \kappa)$$

Definition 13 (BSS algorithm) *Let (f, τ, κ) be a BSS model and $D \subset D_{(\tau,\kappa)} \subset L^1(\Omega, \mathbb{R}^m)$. An ICA algorithm $\iota : D \to \tau$ is then called **BSS algorithm** of (f, τ, κ) if $\iota(X) \circ X$ is independent for every $X \in D$.*

In this case for all $X \in D$ we know that $\iota(X)$ is an ICA of (X, τ, κ).

We speak of linear BSS if both f and τ are linear. We speak of *standard linear BSS* if f is linear and the underlying ICA model is standard linear. Note that affine linear BSS can be easily reduced to linear BSS by centering of all variables.

If we speak of **overcomplete linear BSS** however, we only want f to be linear; since $m < n$ linear elements in τ can only reconstruct m dimensions, not necessary n dimensions.

2.6.1 Uniqueness of standard linear ICA

For standard linear BSS however, all different ICAs of a random vector are equivalent in the sense introduced in section 2.4 given that not more than one source is gaussian. This very important result represents one of the key reasons for the success of linear symmetric BSS, because it says how well BSS algorithms can recover the mixing matrix - essentially they can find it except for scaling and permutation.

Uniqueness results are not so easy in more general mixing models; and if the mixing models get too large, we will see that the set of ICAs gets too large as well meaning that independence is too weak a criterion in order to recover the sources.

If most one of the source variable $S_i := \pi_i \circ S$ is gaussian ($\pi_i \mathbb{R}^n \to \mathbb{R}$ denotes the projection on the i-th coordinate) then for any solution to the quadratic ($m = n$) BSS problem, i.e. any $W \in Gl(n)$ such that $W \circ X$ is independent, W^{-1} is equivalent to A [50]. Vice versa, any matrix $W \in Gl(n)$ such that W^{-1} is equivalent to the mixing matrix A solves the BSS problem taking into account the invariants under scaling and permutation of coordinates.

The following theorem has been shown by Comon [47]; it is a corollary of the Skitovitch-Darmois theorem (see theorem 31 presented in the Appendix B.5). As the undercomplete mixing case (less sources than mixtures, $m < n$) can be easily reduced to the symmetric case ($m = n$), we will state the theorem only for $m = n$ not $m \leq n$ as Comon.

Theorem 6 (Comon) *Let $S \in L^2(\Omega, \mathbb{R}^n)$ be an independent random vector with at most one gaussian component, and $A \in Gl(n)$. Set $X := AS$. Then the following statements are equivalent:*

1. *$X_1, \ldots, X_n$ are pairwise independent.*
2. *X is independent.*
3. *$A \in \Pi(n)$, that is A is the product of a scaling and a permutation matrix.*

We want to call the following corollary to Comon's theorem again a theorem because of its importance in standard linear BSS.

Theorem 7 (Uniqueness of standard linear BSS) *Let $A \in Gl(n)$ and $S \in L^2(\Omega, \mathbb{R}^n)$ be an independent random vector with at most one gaussian component. Then*

$$BSS(S, A, Gl(n), -I) = \Pi(n)A^{-1}.$$

In fact it is easy to generalize this result to an arbitrary number of gaussians in the sources - the mixing matrix is then determined uniquely expect for permutation and scaling on all non-gaussian components. For this denote given an n-dimensional random vector S and $\gamma(S)$ the number of gaussians in S, then we have a similar theorem as the Comon theorem.

Corollary 1 *Let S be an independent n-dimensional random vector S and $A \in Gl(n)$. Let $X := AS$. Then the following statements are equivalent:*

1. *$X_1, \ldots, X_n$ are pairwise independent.*
2. *X is independent.*
3. *$\gamma(X) = \gamma(S)$ and there exists a permutation $P \in Gl(n)$ and $\lambda_i \neq 0$ with*

$$PA = \begin{pmatrix} \lambda_1 & & & * \\ & \cdots & & \cdots \\ & & \lambda_{n-\gamma(S)} & * \\ & 0 & & * \end{pmatrix}.$$

2.7 Identifiability, Separability, and Uniqueness of Nonlinear ICA Models

Main of the result of this paragraph can be found in [195, 193, 65] and traced back to [186, 183].

Classically, linear BSS has been treated most thoroughly. With the growing popularity of ICA, more and more nonlinear algorithms have been proposed, like for example algorithms for postnonlinear ICA [185, 111, 110],

ideas based on various clustering algorithms to approximate the mixing or unmixing models [143, 98] or Almeida's pattern repulsion using density uniformization [123].

A too general analysis model for the ICA leads to a high degeneration thus making independence a too weak criterion. Hence, there are two ways of how to proceed with nonlinear ICA: either use additional properties of the given data like for example time structure [129, 126], or restrict the unmixing model in order to break the degeneracy and achieve convergence in nonlinear ICA algorithms. Since we only want to deal with ICA, we will do the second, which is often called *regularization*.

There are many other nonlinear analysis model, although often not explicitly specified in the literature. Candidates for this are for example self-organizing maps (SOM) and generative topograpic mappings (GTM) [142, 98]. Another unmixing model that only allows *biholomorphic mappings* is used by Hyvärinen and Pajunen [91] in order to give uniqueness results. This model however is problematic as it does not contain the linear model.

By definition, we speak of nonlinear BSS, when both the mixing mapping f and the analysis model τ of the BSS model (f, τ, κ) are nonlinear. Nonlinear models are often either extensions of the linear model, as for example *postnonlinear (PNL) ICA*, which we'll talk about in more detail in the following.

In practice, when recording signals there exist a lot of different sensors, often with nonlinearities such that the mixtures are in principle a linear mixture, only the output is again nonlinearly transformed. This type of mixing model is the basis for *postnonlinear BSS*. For this let $f_1, \ldots, f_n : \mathbb{R} \to (a_i, b_i)$ be n diffeomorphism, and $A \in Gl(n)$. The mixing mapping f_{pnl} then is of the form

$$f_{pnl} = (f_1 \times \ldots \times f_n) \circ A.$$

The unmixing model τ_{pnl} then is of the form

$$\tau_{pnl} := \{B \circ (g_1 \times \ldots \times g_n) | B \in Gl(n), g_i : (a_i, b_i) \to \mathbb{R} \;\; diffeomorphism\}.$$

The nonlinear BSS model $(f_{pnl}, \tau_{pnl}, \kappa)$ is then called the postnonlinear BSS model. Postnonlinear ICA was first introduced by Taleb and Jutten [184, 186] and uniqueness results are known.

For the case of nonlinear mixing environment additional definitions and theorems are required.

Definition 14 *Let $\mathbf{A} \in Gl(n)$ be an invertible matrix, then $\mathbf{A} = (a_{ij})_{i,j=1\ldots n}$ is said to be **absolutely degenerate** if there are two columns $l \neq m$ such that $a_{il}^2 = \lambda a_{im}^2$ for a $\lambda \neq 0$, i.e. the the normalized columns differ only by the signs of the entries.*

Definition 15 *Given a function $\mathbf{f} : U \to \mathbb{R}$ assume there exist $a, b \in \mathbb{R}$ such that at least one is not of absolute value 0 or 1. If $\mathbf{f}(a\mathbf{x}) = b\mathbf{f}(\mathbf{x})$ for all $\mathbf{x} \in U$ with $a\mathbf{x} \in U$; then $\mathbf{f}$ is said to be (a, b)-**homogeneous** or simply **homogeneous**.*

The following lemma characterizing homogeneous functions is from [12]. However we added the correction to exclude the cases $|a|$ or $|b| \in \{0, 1\}$, because in these cases homogeneity does not induce such strong results. This lemma can be generalized to continuously differentiable functions, so the strong assumption of analyticity is not needed

Lemma 6 (Babaie-Zadeh et al. [12]) *Let $f : U \to \mathbb{R}$, be an analytic function that is (a, b)-homogeneous on $[0, \epsilon)$ with $\epsilon > 0$. Then there exist $c \in \mathbb{R}$, $n \in \mathbb{N} \cup \{0\}$ (possibly 0) such that $f(x) = cx^n$ for all $x \in U$.*

Definition 16 (Babaie-Zadeh et al. [12]) *We call a random vector $\mathbf{X}$ with density $p_{\mathbf{X}}(\mathbf{X})$ **bounded**, if its density $p_{\mathbf{X}}(\mathbf{X})$ is **bounded**. Denote $supp\ p_{\mathbf{X}}(\mathbf{X}) := \{\mathbf{x} | p_{\mathbf{X}}(\mathbf{X}) \neq 0\}$ the support of $p_{\mathbf{X}}(\mathbf{X})$ i.e. the closure of the nonzero points of $p_{\mathbf{X}}(\mathbf{X})$.*
*We further call an independent random vector $\mathbf{X}$ **fully bounded**, if $supp\ p_{\mathbf{X}_i}(\mathbf{X}_i)$ is an interval for all i. So we get $supp\ p_{\mathbf{X}}(\mathbf{X}) = [a_1, b_1] \times \cdots \times [a_n, b_n]$.*

Since a connected component of $supp\ p_{\mathbf{X}}(\mathbf{X})$ induces a restricted, fully bounded random vector, without loss of generality we will in the following assume to have fully bounded densities.

Definition 17 (Taleb and Jutten [186]) *A function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is called diagonal or component-wise if each component $f_i(\mathbf{x})$ of $\mathbf{f}(\mathbf{x})$ depends only on the variable x_i.*

In this case we often omit the other variables and write $\mathbf{f}(x_1, \ldots, x_n) = (f_1(x_1), \ldots, f_n(x_n))$ or $\mathbf{f} = f_1 \times \cdots \times f_n$.

Consider now the post-nonlinear blind source separation model in eq. (2.4). We assume the components f_i of $\mathbf{f}$ to be injective analytic functions with nonvanishing derivatives. Then also the f_i^{-1} are analytic.

Post-nonlinear BSS is a generalization of linear BSS, so the indeterminacies of post-nonlinear ICA contain at least the indeterminacies of linear BSS: $\mathbf{A}$ can only be reconstructed up to *scaling* and *permutation*. Here of course additional indeterminacies come into play because of *translation*: f_i can only be recovered up to a constant. Also, if $\mathbf{\Lambda} \in Gl(n)$ is a scaling matrix, then $\mathbf{f}(\mathbf{As}) = (\mathbf{f} \circ \mathbf{\Lambda})((\mathbf{\Lambda^{-1}A})\mathbf{s})$, so $\mathbf{f}$ and $\mathbf{A}$ can interchange scaling factors in each component. Another indeterminacy could occur if $\mathbf{A}$ is not mixing, i.e. at least one observation x_i contains only one source; in this case f_i can obviously not be recovered. For example if $\mathbf{A} = \mathbf{I}$ then $\mathbf{f}(\mathbf{s})$ is already again independent, because independence is invariant under componentwise nonlinear transformation; so $\mathbf{f}$ cannot be found using this method.

A not so obvious indeterminacy occurs if $\mathbf{A}$ is absolutely degenerate. Then only the matrix $\mathbf{A}$ but not the nonlinearities can be recovered by looking at the edges of the support of the fully bounded random vector.

Example 5.

Consider the case $n = 2$, $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 2 & -2 \end{pmatrix}$ and the analytic function $\mathbf{f}(x_1, x_2) = (x_1 + (1/2\pi)sin(\pi x_1), x_2 + (1/\pi)sin(\pi x_2/2))$. Then $\mathbf{A}^{-1} \circ \mathbf{f} \circ \mathbf{A}$ maps $[0,1]^2$ onto $[0,1]^2$. Since both components of $\mathbf{f}$ are injective, we can verify this by looking at the edges.

$$\begin{aligned} \mathbf{f} \circ \mathbf{A}\,(x_1, 0) &= \left(x_1 + \tfrac{1}{2\pi}\sin(\pi x_1), 2x_1 + \tfrac{1}{\pi}\sin(\pi x_1)\right) = \\ &= (1,2)\left(x_1 + \tfrac{1}{2\pi}\sin(\pi x_1)\right), \\ \mathbf{f} \circ \mathbf{A}\,(0, x_2) &= (1,-2)\left(x_2 + \tfrac{1}{2\pi}\sin(\pi x_2)\right), \\ \mathbf{f} \circ \mathbf{A}\,(x_1, 1) &= (1,-2) + (1,2)\left(x_1 - \tfrac{1}{2\pi}\sin(\pi x_1)\right), \\ \mathbf{f} \circ \mathbf{A}\,(1, x_2) &= (1,2) + (1,-2)\left(x_2 - \tfrac{1}{2\pi}\sin(\pi x_2)\right). \end{aligned}$$

So we have constructed a situation in which two uniform sources are mixed by $\mathbf{f} \circ \mathbf{A}$. They can be separated either by $\mathbf{A^{-1}} \circ \mathbf{f^{-1}}$ or by $\mathbf{A}^{-1}$ alone. It is possible to show that the latter also preserves the boundary, although it contains a different post-nonlinearity (namely identity) in contrast to $\mathbf{f}^{-1}$ in the former model. Nonetheless, this is no indeterminacy of the model itself, since $\mathbf{A^{-1}f}(\mathbf{As})$ is obviously not independent. So by looking at the boundary alone, we sometimes cannot detect independence if the whole system is highly symmetric. This is the case if $\mathbf{A}$ is absolutely degenerate. In our example $\mathbf{f}$ is chosen such that the non trivial post-nonlinear mixture looks linear (at the boundary), and this was possible due to the inherent symmetry in $\mathbf{A}$.

If we however assume that $\mathbf{A}$ is mixing and not absolutely degenerate, then we will show for all fully bounded sources $\mathbf{s}$ that except for scaling interchange between $\mathbf{f}$ and $\mathbf{A}$ no more indeterminacies than in the affine linear case exist. Note that if $\mathbf{f}$ is only assumed to be continuously differentiable, then additional indeterminacies come into play.

2.7.1 Separability

In this section it is proven separability of postnonlinear BSS; it will be seen how the two conditions from Definition 2 and Definition 14 turn out to be necessary.

Theorem 8 (Separability of bounded postnonlinear BSS) *Let $\mathbf{A}, \mathbf{W} \in Gl(n)$ and one of them mixing and not absolutely degenerate, $\mathbf{h} : \mathbb{R}^n \to \mathbb{R}^n$ be a diagonal injective analytic function such that $h_i' \neq 0$ and let $\mathbf{s}$ be a fully bounded independent random vector. If $\mathbf{W}(\mathbf{h}(\mathbf{As}))$ is independent, then there exists a scaling $\mathbf{\Lambda} \in Gl(n)$ and $\mathbf{v} \in \mathbb{R}^n$ with $\mathbf{\Lambda A} \sim \mathbf{W}^{-1}$ and $\mathbf{h}(\mathbf{x}) = \mathbf{\Lambda x} + \mathbf{v}$.*

So let $\mathbf{f} \circ \mathbf{A}$ be the mixing model and $\mathbf{W} \circ \mathbf{g}$ the separating model. Putting the two together we get the above mixing-separating model with $\mathbf{h} := \mathbf{g} \circ \mathbf{f}$.

The theorem shows that if the mixing-separating model preserves independence then it is essentially trivial i.e. $\mathbf{h}$ affine linear and the matrices equivalent (up to scaling). As usual, the model is assumed to be invertible, hence identifiability and uniqueness of the model follow from the separability.

2.8 Entropic contrasts for BSS

In blind source separation (BSS), two different separation techniques are mainly used: *Minimal Mutual Information* (*MMI*), where minimization of the mutual output information yields an independent random vector, and *Maximum Entropy* (*ME*), where the output entropy is maximized. However, it is yet unclear why ME should solve the separation problem i.e. result in an independent vector.

Yang and Amari have given a partial confirmation for ME in the linear case in [218], where they prove that under the assumption of vanishing expectation of the sources ME does not change the solutions of MMI except for scaling and permutation.

With reference to Appendix A we can introduce some simple and basic contrasts κ in section 2.5 to solve the BSS problem.

The relation between Mutual Information and Joint Entropy of a random vector $\mathbf{Y}$, can be expressed as (see [55] and Appendix A)

$$I(\mathbf{Y}) = \sum_{i=1}^{N} H\left(Y_i\right) - H(\mathbf{Y}) \tag{2.7}$$

A conceptual description of the eq. (2.7) can be described as followed and is showed for $N = 2$ in Figure 2.5.

Because the equation (2.7) is formed by three terms, one can proceed through three ways to reach the independence of the random variable Y_i. It is possible to minimize the left side $I(\mathbf{Y})$ of eq. (2.7) obtaining the Minimal Mutual Information or MMI approach; it is possible to maximize the joint entropy $H(\mathbf{Y})$ in eq. (2.7) obtaining the Maximum Entropy or ME approach and it is possible to maximize the term $\sum_{i=1}^{N} H\left(Y_i\right)$ obtaining the *Maximum Non-Gaussianity* or (*MNG*) approach.

The graphical representation of the relation between mutual information, marginal entropy and joint entropy is shown in Figure 2.6 (and in Figure A.1 in Appendix A or in [55]).

This figure shows that the independence between Y_1 and Y_2 is maximized when the intersection of the two sets is minimized, ideally vanished. This fact is the same that maximizing the union of the two sets or maximizing the non-covered area of the two single sets. In other words, the

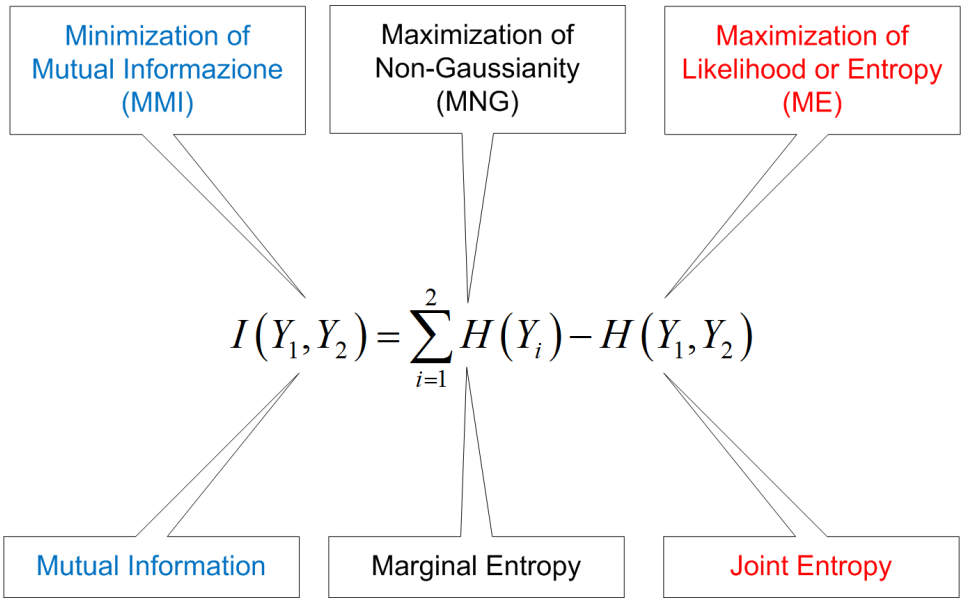

Fig. 2.5: Relation between Mutual Information and Joint Entropy

independence is guaranteed if it is minimized the mutual information of $\mathbf{Y}$ (MMI approach), maximized the joint entropy of $\mathbf{Y}$ (ME approach) or maximized the marginal entropy of Y_1 and Y_2 (MNG approach), respectively. See Figure 2.7 for more details.

The solution points of MMI are kept fixed by ME if no scaling in all layers is allowed. In general, ME however might also change the scaling in the non-output network layers, hence leaving the MMI solution points. So the main question we are trying to answer in this chapter is: When is the entropy a contrast? The following results have been presented in [191, 192, 194, 218].

Bell and Sejnowski [15] propose using the entropy as contrast i.e. maximizing the entropy (ME) of output units of a neural network that is to approach the unmixing mapping. A different contrast however is given by Comon [50]: the mutual information of the output because minimizing the mutual information (MMI) induces statistical independence of the output. This has to be compared with Bell and Sejnowski's suggestion to maximize the entropy of the output. As they note in [15], ME does not always induce MMI and therefore statistical independence. ME performs best when the nonlinear demixing function in the ME algorithm matches with the cumulative distribution of the given source.

However, nowadays a lot of algorithms are based on the ME contrast function like for example extensions of the infomax algorithm [6, 109]. The question then was how this large branch of ME algorithms compares to the MMI methods. For the linear case, Yang and Amari gave a partial answer to that in [218]. They showed that at solution points of ICA determined by MMI,

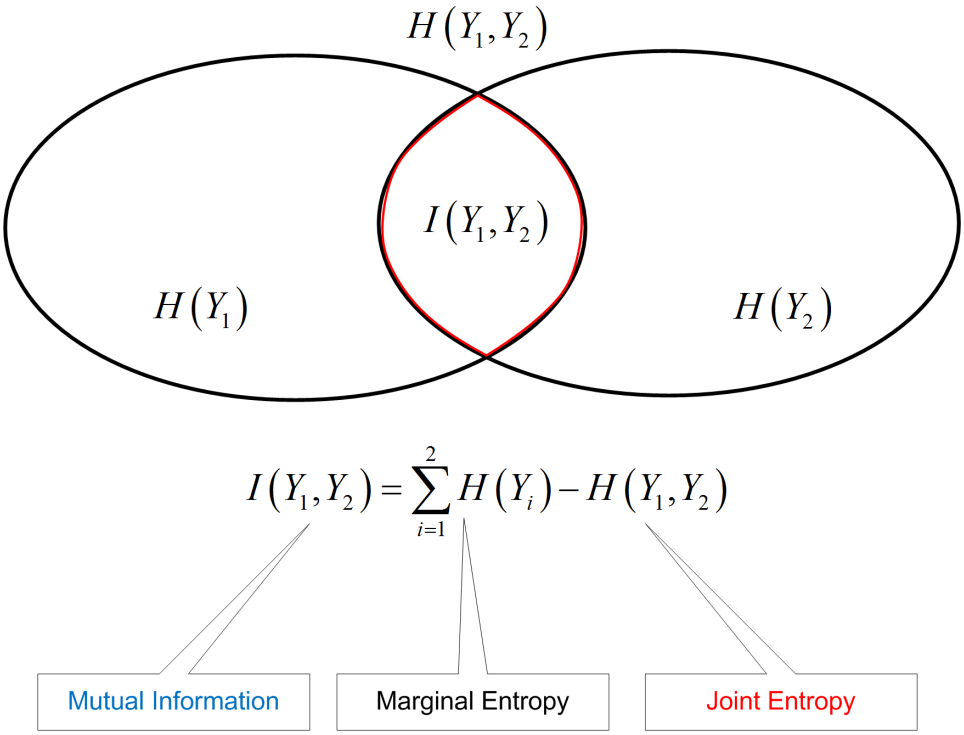

Fig. 2.6: Graphical relation between Mutual Information, Marginal Entropy and Joint Entropy

ME algorithms will only change the demixing matrices by a scaling matrix thus leaving any ME solution unchanged. This result can be interpreted as a local justification for ME, showing that demixing matrices are stable fixed points (except for scaling) of ME algorithms.

In the nonlinear case some algorithms have been proposed (for example [123, 108, 110]), most of them based on the ME contrast function. It is shown in [188, 191] that at certain solution points of MMI, ME will have a local extremum thus only changing the solution in a well-known way (scaling in all weight matrices).

First note that MMI is obviously better suited than ME in terms of finding solutions; ME may terminate at points that do not represent demixing functions. As mentioned above this has for example been shown in [133] and [15].

The basic problem is that in the decomposition of the joint entropy into the difference of the marginal entropies and the mutual information, a transformed random vector with non-zero mutual information might have higher marginal entropy than the sources resulting in a higher joint entropy as well. Therefore even in the linear case [218] it can only be shown that solutions of MMI are also solutions of ME. For this, the well-known

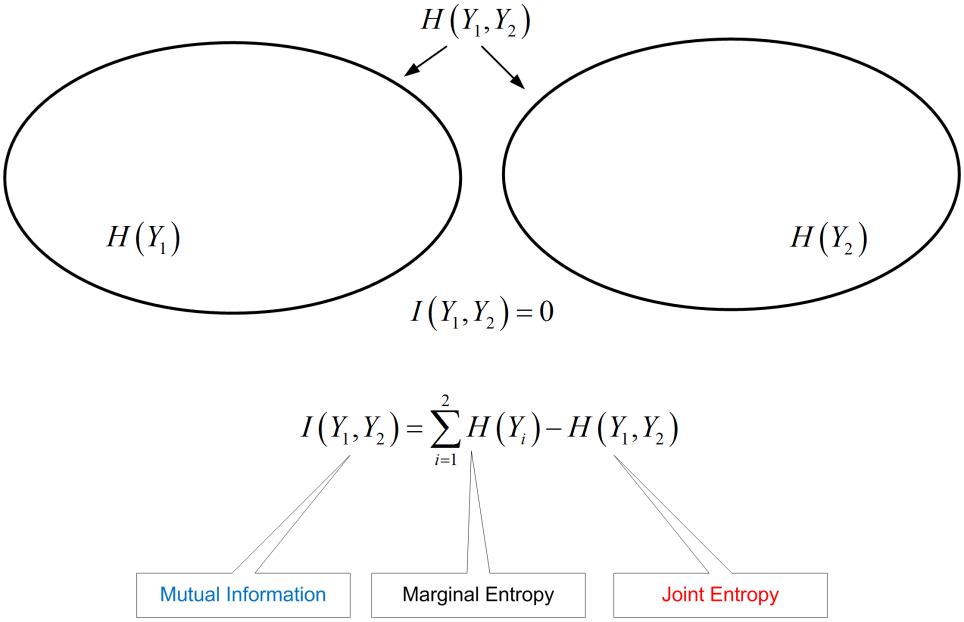

Fig. 2.7: Graphical interpretation of MMI, ME and MNG approaches

uniqueness result by Comon [50] has to be used. Since no such uniqueness results have been found in more general nonlinear settings, it is possible to show only that special demixing functions are solutions of ME.

For a single layer neural netwok Theis has demonstrated in [188, 191] the following

Theorem 9 *Assume the expectation $E(\mathbf{s})$ of the sources vanishes. Let $f \in \tau$ be a scaled solution. If f is a BSS of $(S, \nu, \tau, -I)$, then f is a local extremum of κ_X.*

The theorem says that scaled solutions that are solutions of the BSS problem are local extrema of ME. To be more precise: The ME algorithm transforms scaled solutions that are solutions of BSS into scaled solutions.

3

Why a Complex Model?

—You have never given me a transverse look.
A. Chekhov

THE nature of some problem involves a natural solution in a complex environment [121, 82], due to the need of frequency domain signal processing which is quite common in telecommunication [19] and biomedical applications [29, 30].

Recent breakthroughs in technology and biomedicine have highlighted applications where nonlinearity, non-stationarity, multidimensional data natures, and uncertainty play major roles. Subsequently novel signal processing and machine learning theories have been rapidly developed in order to cater for these new classes of problems.

Apart from the problems with nonlinearity, non-stationarity and noise, novel applications in brain science, communications, and data and sensor fusion are dealing with multidimensional (multichannel) measurements, for which the on-line processing algorithms are essential [120]. If possible (data dimensionality, computational power), the processing of such data should be performed directly in the spaces where the data vectors reside $(\mathbb{R}^n, \mathbb{C}^n)$. Machine learning theories for multidimensional domains are still being developed; this is especially in the presence of nonlinearity and noise.

Following a related study for nonlinear models [1], it is natural to ask ourselves whether it is beneficial to use multidimensional solutions for lower dimensional problems. The simplest, yet extremely important, case to consider is whether complex valued solutions for real valued problems provide a theoretical and performance advantage over standard real valued solutions.

In applications of adaptive systems it is the signal magnitude that is used as the main source of information [141]. Whereas this facilitates the

established algorithms in $\mathbb{R}$, it is important to realize that this way the full information potential (phase information) within the signal is not utilized. Indeed real world processes with the "*intensity*" and "*direction*" component (radar, sonar, vector fields) require the consideration of such phase information [121, 119]. Our question then boils down to whether the processing of such real-valued data in the complex domain $\mathbb{C}$, where the phase information can be accounted for naturally, has advantages over straightforward processing in $\mathbb{R}$. For instance there are some interesting and recent results on the modeling of **wind** profile [71, 72, 119], indicate that the processing in $\mathbb{C}$ (simultaneously speed and direction as a complex vector) has major benefits over the direct processing in (speed only), or in $\mathbb{R}^2$ (speed and direction as independent processes).

3.1 Real against Complex

When it comes to the duality between real and complex processes, two aspects of this duality are highlighted: the importance of phase (direction) information, and the advantages of the simultaneous modeling of the "*intensity*" and "*direction*" component of vector field processes in $\mathbb{C}$ over the so called "*dual univariate*" modeling (where the component of such processes are treated as independent random processes).

Complex valued data are complex either by their nature (e.g. communications) or by convenience of representation (e.g. phasors in circuit theory [43]). Figure 3.1 illustrates such duality between the processing in $\mathbb{R}$ and $\mathbb{C}$. The nature of purely *real* and *complex* signals is obvious: *real* signals are magnitude-only whereas *complex* signals comprise both magnitude and phase components. *Phase only* signals are real signals formed from the phase of a complex signal. *Dual univariate* signals are the components of a complex signal that are processed separately as real valued quantities.

In order to transform a real signal into its complex valued counterpart, it is convenient to make use of the delay or phase associated with the time of arrival of the real valued signal (or vector field) at sensors. When transforming a complex signal into its dual univariate representation, we need to identify the heterogeneous (e.g. wind speed and direction) components within the complex representation. Another interesting scenario is the "*complex to phase only*" transformation for phase only modeling (tracking); this is very practical for cases where the magnitude of the signal has small or no variation [187]. There are a variety of other ways of transforming signals from $\mathbb{R}$ into $\mathbb{C}$ their domain counterparts, most of them are application specific.

Oftentimes in machine learning our aim is to preserve or enhance the phase of a signal(s); this plays major part in perception based modeling (video, speech, haptic, multimodal). While the phase information in 1-D

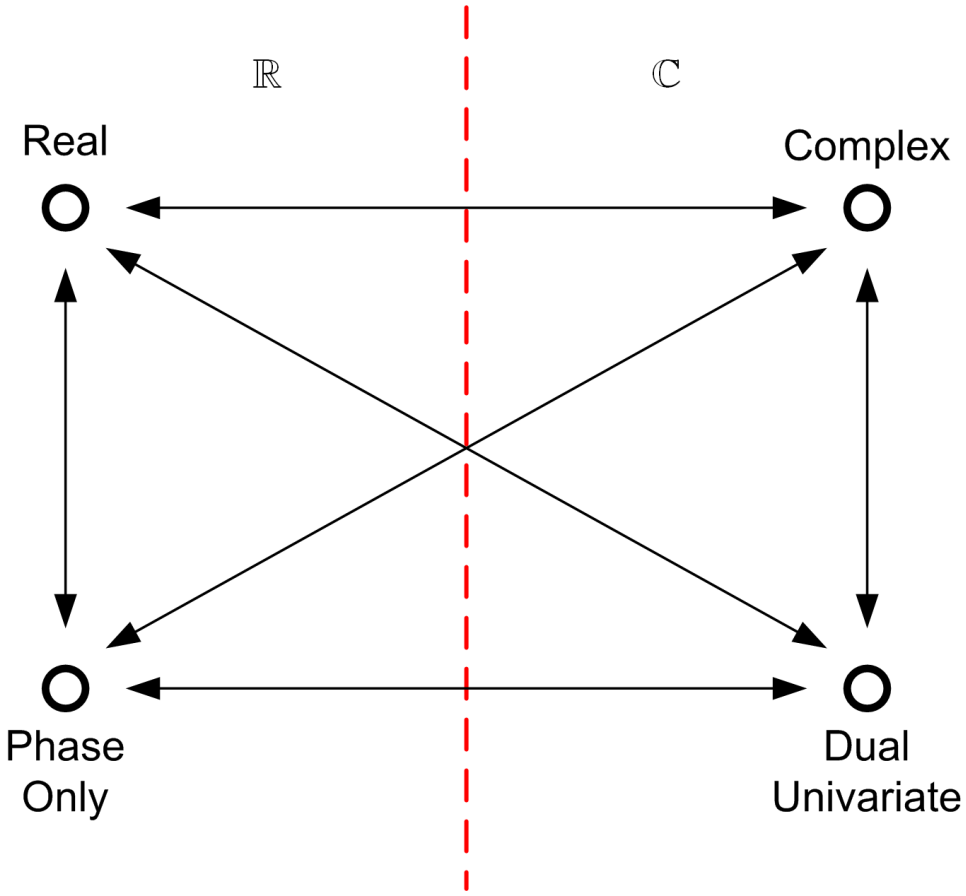

Fig. 3.1: Duality between signal processing in $\mathbb{R}$ and $\mathbb{C}$

signals is subtly hidden, in 2-D signals such as images, the role of the phase of a signal is more obvious [141].

It is therefore natural and convenient to conduct processing of images based solely on their complex valued representation [9]. The subsequent image processing can be now performed directly in $\mathbb{C}$ [82].

3.1.1 Does everyone agree?

In the early 90s several studies on complex random vectors are risen [134, 149, 148] and a great initial excitement pervaded the authors of these works. For example, the *SP Forum* in Signal Processing Magazine (March 1995) defined the work of Bernard Picinbono in [148] as a "*recent extension and refinement of an important concept...*"

Straight after other authors disappointed these works because in their opinion no advances were in such extension in the complex domain. It is the case of Glenn Johonson which had a wide correspondence with Picinbono

in *Signal Processing Magazine* [92, 93, 150].

Johonson does not agree with Picinbono on the usefulness of the complex representation. In fact in [92] he said: "*Engineering is applied science, has a produce and we do not need to carry on "distinction without a difference" debates, either for time series or complex data models. Ideally, authors submitting papers to the IEEE journals have something to contribute to engineering. Contributions include reporting upon techniques that proved effective for achieving an engineering goal, comparing techniques, or explaining in a general context why techniques are effective, and identifying limitations. I see a growing disconnect between scientists and engineers, and the publications of academics, which are too often dubious mathematics. The reference paper supplies only abstract results, and if better examples were used, it would be apparent to nearly all readers that the introduced terms are unnecessary and overblown. The paper invents jargon to compensate for a lack of a consistent definition for the common concept or "complex random number." A complex random number is nothing more than notation or pairs of random numbers, a notation which simplifies many algebraic manipulations. Measurable physical quantities, including those modeled as random variables, lake on values that are real numbers. Imaginary numbers are, well, imaginary. [...]. The complex shorthand is convenient for algebraic manipulation of, for example, analytic signals, but this convenience does not make the results of any measurement complex. Complex random numbers are a well accepted, often convenient, notation for pairs of real random numbers. [...]. A real formulation is the correct formulation for engineering problems, and consequently is the only linear estimate that need be considered.*"

For models exhibiting circularity, the constrained "complex" estimates coincide with the optimal estimates. But, circularity is only a sufficient, and not a necessary condition for the constrained solution to coincide with the optimal solution.

And Johonson continues: "*I am not aware of a single system that is suboptimal because a design engineer mistakingly insisted upon solving the overly constrained, "complex" optimization problem. None or the machinations introduced in the cited reference are required if there is no insistence on making too much of the complex notation for pairs of random variables. If there are notable applications of circularity, [148] chooses not to enlighten us (And I do not count putting already solved problems into new notation.)*"

Therefore Picinbono answered back to Johonson and pointed out the simplicity introduced in formulas by a complex representation. In fact he said in [150]: "*[...] the Laplace transform of the unit step function can be written with two real numbers* $(a^2 + b^2)^{-1}a$ *and* $-(a^2 + b^2)^{-1}b$ *instead of* s^{-1}*, where* $s = a + jb$*. Fortunately, there are many people who think that* s^{-1} *is simpler than its corresponding real and imaginary parts, and this is also my opinion. This justifies the extended use of complex numbers in many fields of pure and applied science. However, there is a risk when doing so, and this especially occurs in estimation problems. I do not know of any book presenting statistical signal processing applied, for example, to spectral analysis or antenna design that does*

not use complex notation. This notation considerably simplifies the presentation. Furthermore, almost all the more mathematically oriented books present estimation theory using complex random variables or processes. [...]. Therefore it is important to know whether or not the circularity assumption is valid for practical problems. One of the purposes of [148] is to study this question and to show for example that circularity has some relationship with stationarity - a very common assumption in random signal problems. However, there are some situations where circularity cannot be taken for granted. Unlike Mr. Johnson, I am aware of some systems using the classical solution where the circularity assumption may not be valid. These systems therefore are not, in general optimum."

And Picinbono continues: "*Curiously, after criticizing the excess of allegedly useless mathematics for engineers, Johnson states that circularity is a "sufficient" condition and, without proof, that it is not "necessary". As a matter of fact, this is right and this point is fully analyzed in [148]. [...]. As an example, let us consider the condition of circularity which implies that the classical solution is also optimal. In complex notation it is simply written by the equation* $E\left\{\mathbf{x}\mathbf{x}^T\right\} = 0$. *It is a property of the complex matrix* $E\left\{\mathbf{x}\mathbf{x}^T\right\}$. *Furthermore it has a very simple physical meaning. Indeed, this condition implies that the vectors* $\mathbf{x}$ *and* $\mathbf{x}exp(j\alpha)$ *have the same second-order statistical properties for any* α, *which is the origin of the term of circularity. In Johnson's' approach, one can write this matrix equation using two real matrix equations. However, we must be consistent in philosophy and ignore the complex vector* $\mathbf{x}$ *itself and only use its real and imaginary parts, say* $\mathbf{x} = \mathbf{x_1} + \mathbf{jx_2}$. *In using these notations the circularity condition* $E\left\{\mathbf{x}\mathbf{x}^T\right\} = 0$ *becomes equivalent to the set of equations*

$$E\left\{\mathbf{x}_1\mathbf{x}_1^T\right\} = E\left\{\mathbf{x}_2\mathbf{x}_2^T\right\} \quad and \quad E\left\{\mathbf{x}_1\mathbf{x}_2^T\right\} = -E\left\{\mathbf{x}_2\mathbf{x}_1^T\right\}$$

It is not clear why anyone would prefer these equations to the single one conveniently written in complex notation. Furthermore, the physical interpretation indicated above does not appear c1carly with real notations and requires a new calculation in which $exp(j\alpha)$ *is decomposed in its real and imaginary parts. Let us also note that ignoring complex notations requires that we write the Fourier analysis with cosine and sine functions instead of complex exponential functions; I do not know of any scientist or engineer that would consider this progress when the analysis is so simple and elegant with complex numbers."*

Consequently Johonson wrote another letter [93], where more moderately he tried to explain the motivation of his previous considerations. In fact he said: "*I was, of course, criticizing [148], and not Gauss', or anyone else's, use of complex notation. For those who have not read my letter [92], I would like to emphasize that I stated that "Complex notation is a well accepted, often convenient, notation for pairs of real random numbers". My point was not to "completely ignore the complex representation" as Picinbono writes [150], but to take issue with the coinage of "circularity" as a valuable concept and to take issue with papers in engineering journals that are of little use to engineers. [...]. These simple facts become*

obscured in the discussions of random vectors exhibiting "circularity." These two well understood cases, uniformly distributed and non-uniformly distributed random phase, have been distorted into "circularity" and "non-circularity", resulting in a sequence of papers. These papers do nothing more than restate the two cases in a more "complex" form. These papers contribute nothing but confusion to already well developed signal processing solutions. I would like to see Picinbono's definition for a complex random variable that requires any modification to conventional formulations of signal processing. [...]. Complex notation is useful. Nevertheless, the point is that, beyond the obvious, there is nothing fundamentally different between real and complex random variables. Picinbono tries to have it both ways, acknowledging that there is no difference on one hand ("...that complex numbers are nothing else but pairs of real numbers ...." [148] and claiming ground-breaking results on the other ("...the estimation theory must be reformulated." [148]. [...]. Why? What is the distinction between a complex vector and "its real and imaginary parts"? (Bring the glass and the water, but not the glass of water?!) My point was that it makes no difference what notation is used-the optimal solution is the optimal solution. What prevents me from freely changing notation between real and complex?"

And Johonson continues: "*I can combine the temperature on top of Mount Everest with the voltage from my wall outlet to form a "complex" signal vector. I can choose a scale for the temperature measurement to make the resulting complex signal "circular". (Asserting that the temperature on Everest is uncorrelated with the voltage of my wall outlet.) What has been added to signal processing by denoting two uncorrelated signals as "circular"? The more widely recognized property of correlation is the more fundamental, and useful, concept. What does inventing jargon for special cases of uncorrelated random vectors add to applications of signal processing? I defy advocates of "circularity" to provide a single example from a prominent signal processing system such as a telephone network, a cellular communication network, a direction finding or spectral estimation based system for which the system would be improved if only the engineers involved had understood "circularity"."*

In other words the complex formulation of a real problem is still an open problem.

3.2 Some examples of scenarios in complex environment

As a matter of fact it is possible to depict three different scenarios as characteristic examples of complex signal processing for BSS problem: namely the separation of sources in convolutive environment, of fMRI mixtures and of telecommunication signals.

3.2.1 Convolutive mixture

A first example where the complex formulation is useful is the blind separation in convolutive environment. As can be seen in Figure 3.2, in closed environment, signals are degraded by the reverber and the distance between the i-th source and the j-th sensor is characterized by the impulsive response $a_{ij}[n]$ [23].

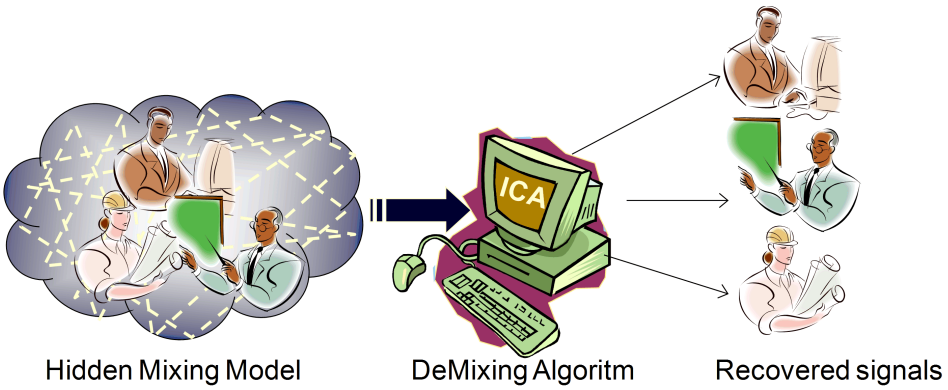

Fig. 3.2: Scenario 1: convolutive mixtures

In this way the j-th mixture can be expressed as

$$x_j[n] = \sum_{i=1}^{N} a_{ij}[n] * s_i[n] = \sum_{i=1}^{N} \sum_{k=0}^{L-1} a_{ij}[k]\, s_i[n-k] \tag{3.1}$$

where $*$ is the convolution operator and L is the length of the impulsive response between the i-th source and the j-th sensor [155].

A more compact matrix formulation of eq. (3.1) is the following one

$$\mathbf{x}[n] = \mathbf{A}[n] * \mathbf{s}[n], \tag{3.2}$$

where

$$\mathbf{A}[n] = \begin{pmatrix} a_{11}[n] & a_{12}[n] & \cdots & a_{1N}[n] \\ a_{21}[n] & a_{22}[n] & \cdots & a_{2N}[n] \\ \vdots & \vdots & \ddots & \cdots \\ a_{N1}[n] & a_{N2}[n] & \cdots & a_{NN}[n] \end{pmatrix}$$

and $a_{ij}[n]$ is the impulsive response defined above. Another way to write the relation (3.1) is to express this mixing process using the *FIR Linear Algebra notation* [107] if the impulsive response $a_{ij}[n]$ is expressed as a FIR filter or a *Moving Average* (*MA*) model. In FIR Linear Algebra, matrices are composed of FIR filters instead of scalars and multiplication between two such FIR matrix elements is defined as their convolution. By implication

the multiplication of two FIR matrices will involve a convolve and accumulate procedure replacing the dot products we would compute for ordinary matrices. In this way eq. (3.2) can be rewritten as $\mathbf{x}[n] = \mathbf{A}[n]\mathbf{s}[n]$.

The aims of BSS is to recover inverse filters $w_{ij}[n]$ such that the signal $u_j[n]$

$$u_j[n] = \sum_{i=1}^{N} w_{ij}[n] * x_i[n] = \sum_{i=1}^{N} \sum_{k=0}^{L-1} w_{ij}[k]\, x_i[n-k] \tag{3.3}$$

is an estimate of the source signal $s_j[n]$.

The compact matrix formulation of eq. (3.3) is the following one

$$\mathbf{u}[n] = \mathbf{W}[n] * \mathbf{x}[n], \tag{3.4}$$

where

$$\mathbf{W}[n] = \begin{pmatrix} w_{11}[n] & w_{12}[n] & \cdots & w_{1N}[n] \\ w_{21}[n] & w_{22}[n] & \cdots & w_{2N}[n] \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1}[n] & w_{N2}[n] & \cdots & w_{NN}[n] \end{pmatrix}$$

and $w_{ij}[n]$ is an estimate of the inverse impulsive response $a_{ij}[n]$ defined above.

Solving the ICA problem (3.2) in the time domain is more complex than the instantaneous one (2.1) because it is necessary to recovery a set of impulsive responses. A more efficient solution can be obtained in the frequency domain using the convolution theorem [155]: *the convolution operator in the time domain becomes a product operator in the frequency domain*.

Using a *short-time discrete Fourier transform* (*STFT*), the mixing model (3.2) is approximated as [174, 118, 16]

$$\mathbf{x}(f,\tau) = \mathbf{A}(f)\mathbf{s}(f,\tau), \tag{3.5}$$

where f denotes the frequency bin, τ is the frame index, $\mathbf{s}(f,\tau)$ and $\mathbf{x}(f,\tau)$ are the short-time Fourier transform of the source vector $\mathbf{s}[n]$ and mixture vector $\mathbf{x}[n]$ respectively. The separation process can be formulated in each frequency bin as

$$\mathbf{u}(f,\tau) = \mathbf{W}(f)\mathbf{x}(f,\tau). \tag{3.6}$$

Eqs. (3.5) and (3.6) show that the solution of ICA problem in frequency domain involves a solution of f_{max} instantaneous ICA problem, where f_{max} is the number of frequency bins used (see Figure 3.3). Moreover the short-time Fourier transform of the source vector $\mathbf{s}[n]$ and mixture vector $\mathbf{x}[n]$ are complex-valued. So the vectors $\mathbf{s}(f,\tau)$ and $\mathbf{x}(f,\tau)$ are complex vector. This justify the use of *complex ICA* [23].

In frequency-domain BSS, the scaling and permutation problem occurs (see eq. (2.3)), i.e., the estimated source signal components are recovered with a different order and gain in the different frequency bins. The scaling

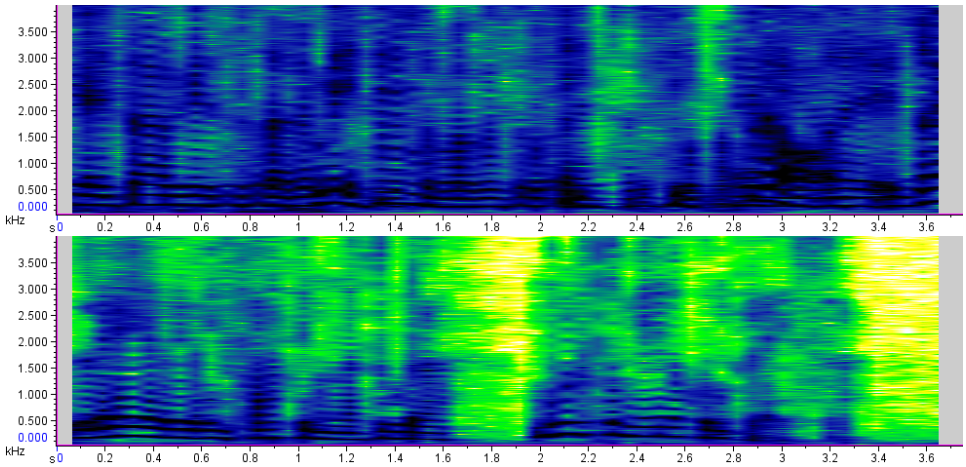

Fig. 3.3: An example of a spectrogram of two mixtures

invariance means that the scaling of every frequency bin can be different, which will of course result in spectral deformation of the original sounds. This problem can be remedied by forcing the determinant of the de-mixing matrices to unity. The permutation invariance is a more difficult problem which is still open to a satisfying solution. This problem is not evident in all cases. For simple mixing filters all de-mixing matrices converge to the same permutation. Using more complicated filters problems might rise. In such cases careful selection of the adaptation parameters in addition to decaying learning rate and momentum are very helpful since they can eliminate random permutation changes in training and preserve the same permutation throughout. Several solutions to permutation ambiguity were proposed [118, 80], one powerful of them was that based on *Direction Of Arrivals* (*DOA*) estimation [130, 131].

3.2.2 Functional magnetic resonance

A second example is the separation of signals recovered by the *Functional Magnetic Resonance Imaging* or *fMRI*. fMRI is a noninvasive, powerful tool that has been utilized in both research and clinical arenas since the early 1990s [138] and has provided valuable insights to the understanding of the human brain function. fMRI has enabled researchers to directly study the temporal and spatial changes in the brain as a function of various stimuli. Because it relies on the detection of small intensity changes over time, fMRI poses significant challenges for data analysis techniques. Traditional model-based analysis approaches - such as linear regression - are robust, yet often too rigid to capture the richness of the human brain activation. Independent component analysis (ICA), on the other hand, is a *datacentric* approach that provides a more flexible framework for the analysis of fMRI data.

For simplicity and tractability, most fMRI analysis techniques to date have discarded the phase of the fMRI data. However, the phase information may be quite valuable for the analysis of the natively complex fMRI data. ICA facilitates the analysis of fMRI data in its complex form by eliminating the need to explicitly model the phase behavior. In what follows, we discuss the application of real and complex-valued ICA to fMRI data analysis and present two examples where ICA has proved particularly useful.

The main advantage of the ICA is the ability to model cognitive processes for which detailed a priori models of brain activity are not available. We present two such examples. The first example considers a paradigm that involves simulated driving, i.e., study of a naturalistic behavior, that is very difficult to model using standard regression-based approaches [132]. The second example demonstrates the application of ICA to fMRI data in its native, complex-valued form. It is particularly difficult to model complex fMRI response using regression - based methods given that very little is known about the phase characteristics of fMRI data.

Processing the fMRI data in its native complex form is attractive for a number of reasons. Using both the magnitude and phase data is expected to:

- increase sensitivity in the identified components;
- elucidate different brain networks while studying the brain connectivity, as the magnitude and phase potentially activate different areas can be stronger when the phase and magnitude of the data are jointly processed;
- localize better the origin of the signal by helping distinguish larger vessels from smaller ones [221].

Most fMRI studies involve a *neurobehavioral* paradigm in which a participant is exposed to sensory stimuli and asked to perform a set of mental and/or motor tasks. A given volume is then collected through slices within a given repetition time, which is usually on the order of a few seconds. The acquired data set includes a brain volume movie with a temporal resolution specified by the time of repetition.

The MRI signal is acquired as a quadrature signal using two orthogonal detectors as shown in Figure 3.4. The signal that is acquired in the complex frequency space (*k-space*) is inverse Fourier transformed into the complex image space. From this point on, almost all fMRI studies analyze only the magnitude images from the MRI scanner, since these are better understood, and discard the information contained in the phase images.

The data are analyzed to determine the *voxels* with significant temporal signal change, which are then super-thresholded and overlaid on an anatomical image. The volume data is then organized into a matrix $\mathbf{X}$ such that each

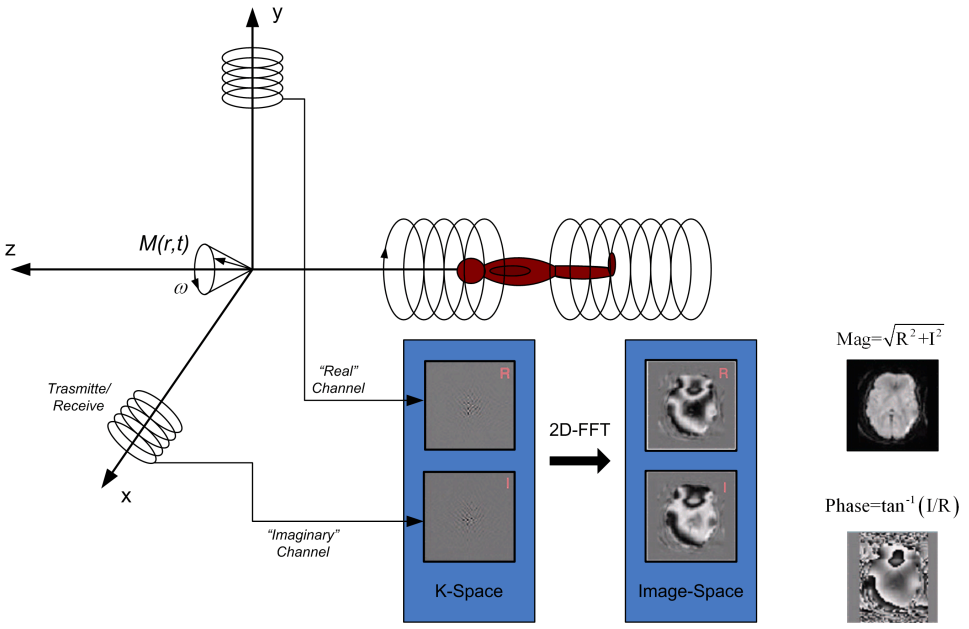

Fig. 3.4: Scenario 2: functional magnetic resonance

row is formed by concatenation of the slices at a given time instant resulting in the $T \times V$ matrix. ICA can be easily applied to cases when temporal dynamics are not well understood, or are not available.

When applied to fMRI data, ICA assumes that there exists a set of non-systematically overlapping (spatially independent) brain networks with their associated time courses. This is referred to as spatial ICA and it is the most popular form of ICA for fMRI data. Alternatively, one can look for temporally independent sources, in which case the sources will correspond to the time courses.

ICA approaches that rely on nonlinear functions to implicitly generate the higher-order statistics (HOS) to achieve independence offer practical and effective solutions to the ICA problem. Two such popular approaches are based on maximum likelihood (ML) - which is equivalent to information maximization - and *maximization of negentropy* (*MN*) [88]. It can be shown that the two measures are equivalent when the mixing matrix is constrained to be unitary. The developments in [3] and [136] present complex ICA algorithms based on ML and MN, respectively.

There are several types of signals that can be encoded within the hemodynamic signals measured by fMRI. Some of these were identified by McKeown in the first application of ICA to fMRI [127].

In general, fMRI data may be grouped into *signals of interest* and *signals not of interest* [25, 27]. The signals of interest include task-related, function-related, and transiently task-related. The task-related signal has already

been mentioned and is the easiest to model. A reference waveform, based upon the paradigm, is correlated with the data. The responses of the brain to a given task may not be regular however, for example the signal may die out before the stimulation is turned off or change over time as repeated stimuli are applied, leading to a transiently task-related signal. It is also conceivable that there are several different types of transiently task-related signals coming from different regions of the brain. The function-related signal manifests as similarities between voxels within a particular functional domain (e.g., the motor cortex on one side of the brain will correlate most highly with voxels in the motor cortex on the opposite side of the brain). An exciting application of this is for identifying synchronous auditory cortex activity [28]. Most of these fMRI signals have been examined with ICA and other methods and have been found to be sub-Gaussian in nature (except perhaps the artifacts mentioned in the next section).

The signals not of interest include physiology-related, motion-related, and scanner-related signals. Physiology-related signals such as breathing and heart rate tend to come from the brain ventricles (fluid filled regions of the brain) and areas with large blood vessels present, respectively. Motion-related signals can also be present and tend to be changes across large regions of the image (particularly at the edges of images).

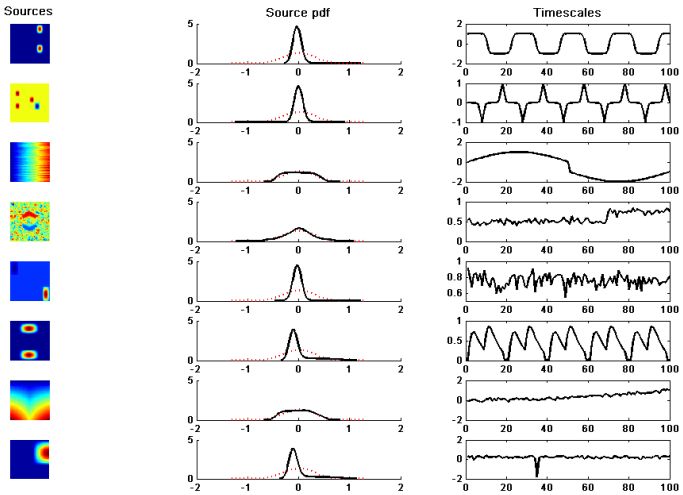

Fig. 3.5: Original data for fMRI

Using the basic knowledge of the statistical characteristics of the underlying sources, it is possible to simulate fMRI-like source images. In Figure 3.5 it is shown a typical data set of fMRI data, containing 5 super-gaussian sources, a gaussian source and 2 sub-gaussian source. Each simulated source is a 60×60 image with 100-point time course. The image is reshaped into

vectors by concatenating columns of the image matrix. This simulation was done with the GIFT toolbox [53]. GIFT is a MATLAB-based ICA/BSS tool that includes a number of analysis and visualization tools in a user friendly graphical interface. The signals $\mathbf{x}$ observed by the detectors are considered to be a mixture of these 8 source signals $\mathbf{s}$

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n]. \tag{3.7}$$

which is formally identical to eq. (2.1).

3.2.3 Telecommunications

A third useful example is represented by telecommunication: the transmitted (*band-pass*) signals are represented by the *complex envelope* $\underline{\mathbf{s}}[n] = \mathbf{s}_R[n] + j\mathbf{s}_I[n]$ which is a complex signal. A sample scenario is depicted in Figure 3.6.

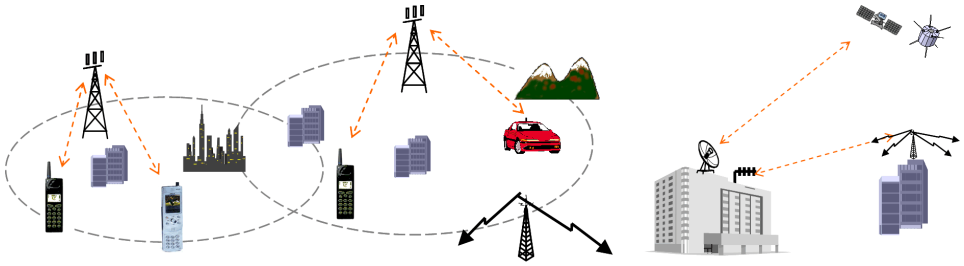

Fig. 3.6: Scenario 3: telecommunications

Consider a band-pass signal $s(t)$ whose Fourier Transform $S(f)$ is non-negligible only in a *band* of frequencies of total extent $2B$, say, centered about some frequency $\pm f_c$. We refer to f_c as the *carrier frequency*. In the majority of communication signals, we find that the *bandwidth* $2B$ is small compared with f_c and so we refer to such a signal as a *narrow-band* signal.

Let $s^+(t)$ be the *analytic signal* [155], then the *complex envelope* $\underline{s}(t)$ is defined as

$$\underline{s}(t) = s^+(t)e^{-j2\pi f_c t} \tag{3.8}$$

Applying the frequency-shifting property of the Fourier Transform to eq. (3.8) we find that the spectrum of the complex envelope $\underline{s}(t)$ is limited to the band $-B \leq f \leq B$ and centered at the origin. That is, the complex envelope $\underline{s}(t)$ of a band-pass signal $s(t)$ is a *low-pass* signal, which is an important result.

We may thus express the original band-pass signal $s(t)$ in terms of the complex envelope $\underline{s}(t)$ as follows:

$$s(t) = Re\left\{\underline{s}(t)exp(j2\pi f_c t)\right\} \tag{3.9}$$

In general, $\underline{s}(t)$ is a complex-valued quantity; to emphasize this property, we may express it in the form

$$\underline{s}(t) = s_R(t) + j s_I(t) \tag{3.10}$$

where $s_R(t)$ and $g_I(t)$ are both real-valued low-pass functions; their low-pass property is inherited from the complex envelope $\underline{s}(t)$. We may therefore use Equations (3.9) and (3.10) to express the original band-pass signal $S(t)$ in the canonical, or standard, form:

$$s\left(t\right) = s_R\left(t\right)\cos\left(2\pi f_c t\right) - s_I\left(t\right)\sin\left(2\pi f_c t\right) \tag{3.11}$$

We refer to $s_R(t)$ as the *in-phase component* of the band-pass signal $s(t)$ and to $s_I(t)$ as the *quadrature component* of the signal.

According to eq. (3.10), the complex envelope $\underline{s}(t)$ may be pictured as a *time-varying phasor* positioned at the origin of the (s_R, s_I)-plane. With time t varying, the end of the phasor moves about in the plane.

The effect of the channel and the *multi-path effect* on the *multiuser* transmitted signal $\mathbf{s} = [s_1, \ldots, s_N]^T$ is the received signal $\mathbf{x} = [x_1, \ldots, x_N]^T$ which consists in a mixture or a convolutive mixture of the original signals $\mathbf{s} = [s_1, \ldots, s_N]^T$. In this sense the model of the received signal is

$$\mathbf{x}\left[n\right] = \mathbf{A}\mathbf{s}\left[n\right] \tag{3.12}$$

which is formally identical to eq. (2.1).

Moreover in digital radio links, since on-board satellite power is a precious resource, to have an high efficiency, the transmitter *high-power amplifier* (*HPA*) operates near the saturation point: nonlinearities are introduced that can cause serious performance degradation of the received signal [156]. An equalization step is the needed [19, 199].

The nonlinearity of a typical HPA, a *traveling-wave tube* (TWT) or a *GaAs FET amplifier*, affects both amplitude (AM/AM conversion) and phase (AM/PM conversion) of the amplified signal, and can be considered as *memoryless*, i.e. the HPA is a nonlinear system without memory under a wide range of operational conditions [161]. However, in practice, the transmitter contains a pulse shaping circuit (*modulator*) at the baseband or at the *intermediate frequency* (*IF*) stage virtually in all digital radio systems. Therefore, the overall baseband-equivalent system (the cascade of transmitter, HPA nonlinearity and receiver) is a *nonlinear* system with memory. In this sense the received signal can be thought as a nonlinear mixture, expressed as in eq. (2.4) by

$$\mathbf{x}\left[n\right] = \mathbf{F}\left(\mathbf{A}\mathbf{s}\left[n\right]\right) \tag{3.13}$$

4

BSS and ICA in Complex Environment

—I was born not knowing and
have had only a little time to
change that here and there.
R. Feynman

THE aim of this section is to extend the BSS problem introduced in previous chapter 2 to the complex domain. In particular the condition of identifiability, separability and uniqueness are extended to the complex environmente.

4.1 Introduction

Let us consider a vector $\mathbf{s}[n] = [s_1[n], \ldots, s_N[n]]^T$ of N complex sources at time n ($\mathbf{s}[n] \in \mathbb{C}^N$). The k-th source can be expressed as $s_k[n] = s_{Rk}[n] + js_{Ik}[n]$, where s_{Rk} and s_{Ik} are the real and imaginary parts of the k-th complex-valued source signal and $j = \sqrt{-1}$ is the imaginary unit. The goal of *complex BSS* is to recover the complex signal $\mathbf{s}[n]$ from observations of the complex mixture $\mathbf{x}[n] = [x_1[n], \ldots, x_N[n]]^T$, where the k-th mixture can be expressed as $x_k[n] = x_{Rk}[n] + jx_{Ik}[n]$, x_{Rk} and x_{Ik} are its real and imaginary part. In this way the model in eqs. (2.1) and (2.2) are still valid but the mixing matrix $\mathbf{A}$ and the de-mixing matrix $\mathbf{W}$ are complex matrices ($a_{ij} \in \mathbb{C}$ and $w_{ij} \in \mathbb{C}$):

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] \tag{4.1}$$

for the mixing model and

$$\mathbf{u}[n] = \mathbf{W}\mathbf{x}[n] \tag{4.2}$$

for the de-mixing model (see Figure 4.1).

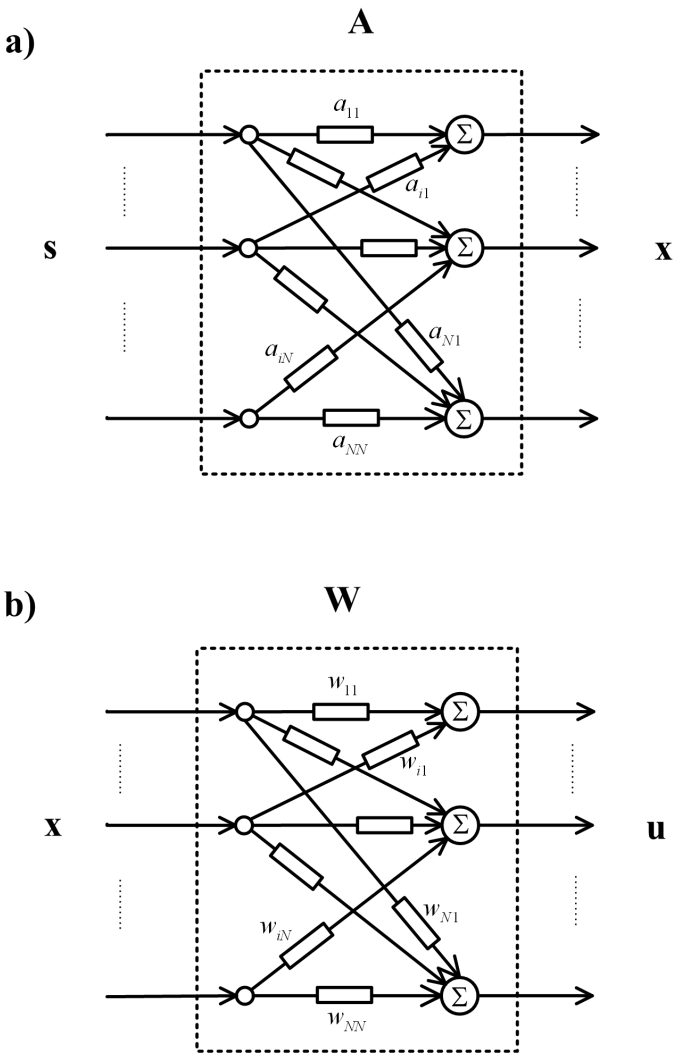

Fig. 4.1: The complex mixing (up) and de-mixing (down) models

In similar way we can describe the case of PNL mixtures where the nonlinear function involved in the models are complex functions too:

$$\mathbf{x}[n] = \mathbf{x}_R[n] + j\mathbf{x}_I[n] = \mathbf{F}(\mathbf{v}[n]) \tag{4.3}$$

where $\mathbf{v}[n] = \mathbf{A}\mathbf{s}[n]$, the nonlinear function $\mathbf{F}(\mathbf{v}[n])$ is the model of the nonlinear distortion in the complex domain and $\mathbf{F}(\mathbf{v}[n]) = [f_1(v_1[n]), \ldots, f_N(v_N[n])]^T$, where $f_k(v_k[n])$ is the k-th complex nonlinear distorting function. For the mirror de-mixing model:

$$\mathbf{u}[n] = \mathbf{u}_R[n] + j\mathbf{u}_I[n] = \mathbf{W}\mathbf{r}[n] = \mathbf{W} \cdot \mathbf{G}(\mathbf{x}[n]) \tag{4.4}$$

where $\mathbf{r}[n] = \mathbf{G}(\mathbf{x}[n])$, the nonlinear function $\mathbf{G}(\mathbf{x})$ is the model of nonlinear compensating function in the complex domain and $\mathbf{G}(\mathbf{x}[n]) = [g_1(x_1[n]),$

$\ldots, g_N(x_N[n])]^T$, where $g_k\left(x_k\left[n\right]\right)$ is the k-th complex nonlinear compensating function (see Figure 4.2).

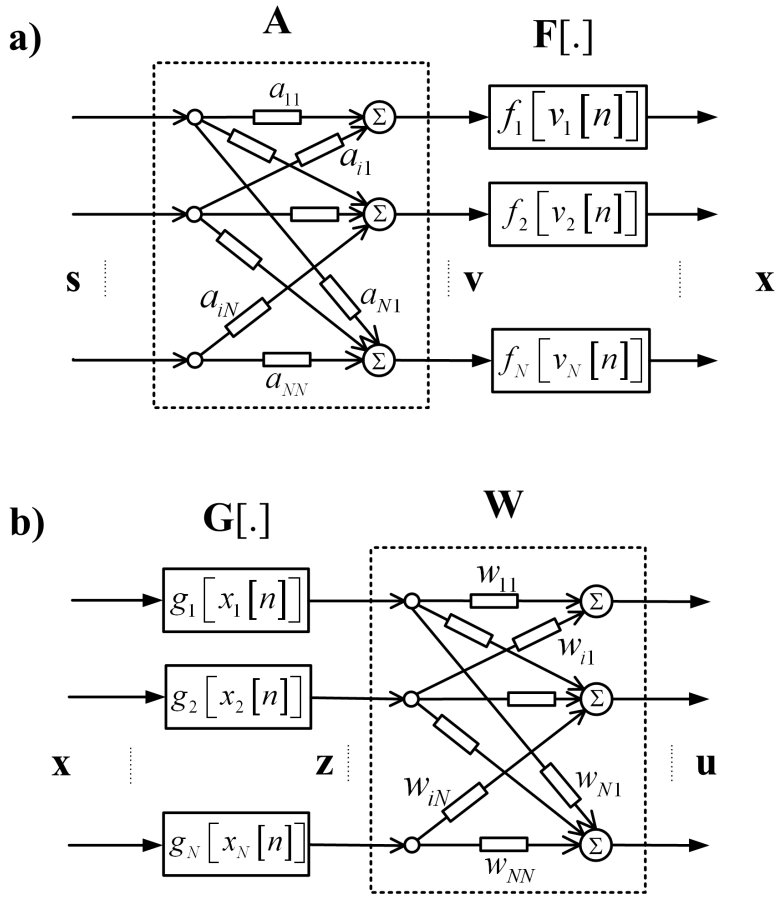

Fig. 4.2: The complex PNL model: mixing model (up) and de-mixing mirror model (down)

The separability, identifiability and uniqueness of the solution is demonstrated in [189] and [66] in the linear and instantaneous case, while the PNL mixtures is not treated yet in literature. It is intention of this thesis to extend these results in the nonlinear environment.

4.2 Identifiability, separability and uniqueness of linear complex ICA

Referring to the model (4.1) and the topics in Appendix B, the couple $(\mathbf{A}, \mathbf{s})$ is called a *representation* of r.vc. $\mathbf{x}$. If no column in the mixing matrix $\mathbf{A}$ is *collinear* with another column in the matrix, i.e., all columns are pairwise linearly independent, the representation is called *reduced*. All representations

are assumed to be reduced throughout this paragraph. Furthermore, a reduced representation for the r.vc. $\mathbf{x}$ in the model (4.1) is called proper, if it satisfies all the assumptions made about the model. Main of the results in this section are taken by [66, 63].

The model of (4.1) is defined to be

1. *identifiable*, or the mixing matrix is (essentially) unique, if in every proper representations $(\mathbf{A}, \mathbf{s})$ and $(\mathbf{B}, \mathbf{r})$ of $\mathbf{x}$, every column of complex matrix $\mathbf{A}$ is collinear with a column of complex matrix $\mathbf{B}$ and vice versa;

2. *unique* if the model is identifiable and furthermore the source r.vc.s $\mathbf{s}$ and $\mathbf{r}$ in different proper representations have the same distribution for some permutation up to changes of location and complex scale;

3. *separable*, if for every complex matrix $\mathbf{W}$ such that $\mathbf{Wx}$ has m independent components, we have $\mathbf{\Lambda P s} = \mathbf{Wx}$ for some diagonal matrix $\mathbf{\Lambda}$ with nonzero diagonals and permutation matrix $\mathbf{P}$. Moreover, such a matrix $\mathbf{W}$ has to always exist.

It is completely possible for the model (4.1) to be identifiable but not unique nor separable as it is shown in the next example.

Example 6.

As an example of a model which is identifiable but is not separable nor unique, consider independent non-normal r.v.s s_k, $k = 1, \ldots, 4$. Let η_1, η_2, and η_3 be independent standard normal r.v.s with the same *circularity coefficient* (see Definition 39). Then also r.v.s $\eta_1 + \eta_2$ and $\eta_1 - \eta_2$ are independent. Now

$$\begin{pmatrix} s_1 + s_3 + s_4 + \eta_1 + \eta_2 \\ s_2 + s_3 - s_4 + \eta_1 - \eta_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 + \eta_1 \\ s_4 + \eta_2 \end{pmatrix} =$$
$$= \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 + \eta_1 + \eta_2 \\ s_2 + \eta_1 - \eta_2 \\ s_3 \\ s_4 \end{pmatrix}$$

which shows that the corresponding model can not be unique. However, it is identifiable. R.v.s of the form $\mathbf{s} + \mathbf{n}$, where $\mathbf{n}$ is a normal r.v. independent of , are said to have a *normal component*.

It follows from the reduction assumption that the number of columns, i.e., the number of sources or the *model order*, is the same in every proper representation of $\mathbf{x}$ in identifiable models. If $\mathbf{W}$ is a separating matrix, then linear manifolds of $\mathbf{\Lambda P}$ and $\mathbf{W}$ must coincide, and therefore $p \geq rank[\mathbf{W}] =$

$rank[\mathbf{\Lambda P}] = m$, i.e., there has to be at least as many mixtures as sources in a separable model. This fact also emphasizes that identifiability of the model (4.1) depends also on the linear operator structure, and since the linear operators defined on $\mathbb{R}^{2n}$ and $\mathbb{C}^n$ are not isomorphic, one can not simply consider real-valued model with twice the observation dimension when studying the complex ICA model (4.1). This is illustrated in the following example.

Example 7.

By simply considering real-valued models with twice the dimension, it may actually seem that the complex separation is possible only under very strict conditions. Indeed, let $r_k, k = 1, \ldots, 4$ be independent real-valued r.v.s, and let $\mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}$, and $\mathbf{B_2}$ be 2×2 nonsingular real matrices. Define $s_1 = \mathbf{A}(r_1\ r_2)^T$ and $s_2 = \mathbf{A}(r_3\ r_4)^T$. Now s_1 and s_2 are independent, but so are also y_1 and y_2

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{B}_2 \end{pmatrix} \mathbf{P} \begin{pmatrix} \mathbf{A}_1^{-1} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{A}_2^{-1} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

for any permutation matrix $\mathbf{P}$. However, y_1 and y_2 are mixtures of s_1 and s_2 for many permutations $\mathbf{P}$.

The previous example is easily generalized to the ICA models that have *multidimensional* independent sources, i.e., one is looking for independent multidimensional subspaces. The example shows that such models can not be identified or separated without additional constraints on the internal dependency structure of the sources or the allowed mixing matrices.

Since linear operators in complex and real spaces are not isomorphic, the classes of separable source r.v.s are not the same. That is, some source r.v.s considered in complex mixtures can be separated although their real-valued representations in real mixtures can not. This is shown in the next example.

Example 8.

Let $\eta_1, \ldots, \eta_{2m}$ be independent standard zero mean unit variance real Gaussian r.v.s. Define

$$\eta = \left(\frac{1}{\sqrt{m+1}} \left(\sqrt{m}\eta_1 + j\eta_{m+1} \right), \frac{1}{\sqrt{m}} \left(\sqrt{m-1}\eta_2 + j\eta_{m+2} \right), \ldots, \frac{1}{\sqrt{2}} \left(\eta_m + j\eta_{2m} \right) \right)^T$$

Now it is easily seen that η is a standard normal r.vc. with the distinct circularity spectrum $\lambda[\eta] = (\frac{m-1}{m+1}, \frac{m-2}{m}, \ldots, 0)^T$. If $\eta_{\mathbb{R}}$ is taken as the source r.vc. in the real-valued ICA model, i.e., $\mathbf{y} = \mathbf{B}\eta_{\mathbb{R}}$ and $\mathbf{B}$ is a $2n \times 2m$ real-valued matrix, $n \geq m$, the model is not separable [64]. However, the complex model involving η itself, i.e., $\mathbf{x} = \mathbf{A}\eta$ and $\mathbf{A}$ is a $n \times m$ complex-valued matrix, is separable by Corollary 3.

The following characterization theorem is the base of the identifiability and uniqueness theorems. It is an extension of a real Theorem 1 [64] to the complex case. The idea of the proof is similar to the proof of Darmois-Skitovich theorem [188].

Theorem 10 *Let* $(\mathbf{A}, \mathbf{s})$ *and* $(\mathbf{B}, \mathbf{r})$ *be two reduced representations of a* n*-dimensional complex r.vc.* $\mathbf{x}$*, where* $\mathbf{A}$ *and* $\mathbf{B}$ *are constant complex matrices of dimensions* $n \times m$ *and* $n \times q$*, respectively, and and are complex r.vc.s with independent components. Then the following properties hold:*

1. *if the* k*-th column of* $\mathbf{A}$ *is not collinear with any column of* $\mathbf{B}$*, then the r.v.* s_k *is complex normal;*

2. *if the* k*-th column of* $\mathbf{A}$ *is collinear with the* l*-th column of* $\mathbf{B}$*, then the logarithms of the cf.s of r.v.s* s_k *and* r_k *differ by a wide sense polynomial in a neighborhood of the origin.*

4.2.1 Separability

ICA is commonly used as a Blind Source Separation method, where the problem is to *extract* the original signals from the observed linear mixture. Therefore, separability of the ICA model is an important issue. The separability theorem for the complex ICA model below may be surprising, since it allows also separation of some complex normal mixtures.

Theorem 11 *The model of* (4.1) *is separable if and only if the complex mixing matrix* $\mathbf{A}$ *is of full column rank and there are no two complex normal source r.v.s with the same circularity coefficient.*

If the source $\mathbf{s}$ has finite second order statistics and the circularity spectrum $\lambda[\mathbf{s}]$ is distinct, then the separation can be achieved by simply performing the strong-uncorrelating transform by Corollary 3. In this case, there is no additional restrictions on the distribution of the source r.v.s, and therefore some *normal* r.v.s can be also separated. An example of such a mixture is seen in Example 9.

4.2.2 Identifiability

Identifiability considers reconstruction of the mixing matrix. This is useful in some problems, where the immediate interest may not be in the sources themselves but in how they were mixed (e.g., channel matrix in MIMO communications).

Theorem 12 *The model of* (4.1) *is identifiable, if*

1. *no source r.v. is complex normal, or*

2. **A** *is of full-column rank and there are no two complex normal source r.v.s with the same circularity coefficient.*

There is a striking contrast between the two cases in Theorem 12. Namely, if there are more sources than mixtures not a single normal r.v. is allowed whereas in the other case all source r.v.s can be normal. The following example shows the reason why we can not allow a single normal r.v. for identifiability when there are more sources than sensors.

Example 9.
Consider independent non-normal r.v.s s_1, s_2, and standard normal r.v.s η_1 and η_2 with the same circularity coefficient. Now

$$\mathbf{x} = \begin{pmatrix} s_1 + s_2 + 2\eta_1 \\ s_1 + 2\eta_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 + 2\eta_1 \\ 2\eta_2 \end{pmatrix} =$$
$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} s_1 + \eta_1 + \eta_2 \\ s_2 \\ \eta_1 - \eta_2 \end{pmatrix}$$

and the last column shows that the model is not identifiable.

It is evident from the previous example and from the separation theorem that another identifiability condition could be formulated by essentially allowing a single normal r.v. and not allowing other source r.v.s to have normal components with the same circularity coefficient. However, this condition is unnecessarily complicated. Therefore, it is not stated in a formal manner.

4.2.3 Uniqueness

Uniqueness considers the case where one is interested not only in the mixing matrix but also in the distribution of the sources.

Theorem 13 *The model of* (4.1) *is unique if either of the following properties hold.*

1. *The model is separable.*
2. *All cf.s of source r.v.s are analytic (or all cf.s are non-vanishing), and none of the cf.s has an exponential factor with a wide sense polynomial of degree at least two, i.e., no source r.v. has the cf. φ such that $\varphi(z) = \varphi_1(z)exp(\mathcal{P}(z, z^*))$ for a cf. $\varphi_1(z)$ and for some wide sense polynomial $\mathcal{P}(z, z^*)$ of degree at least two.*

A non-unique but identifiable mixture was described in Example 7. By slightly restricting the allowed mixing matrices, it is possible in the real case to obtain more classes of unique models [64].

4.3 Identifiability, separability and uniqueness of nonlinear complex ICA

The results on the identifiability, separability and uniqueness of the solution in the case of post nonlinear (PNL) mixtures can be extended to the complex domain too.

For $m, n \in \mathbb{N}$ let $Mat(m \times n; \mathbb{C})$ be the space of complex $m \times n$ matrices, and

$$Gl\,(n; \mathbb{C}) = \{W \in Mat\,(n \times n; \mathbb{C}) \mid \det\,(W) \neq 0\,\}$$

be the general linear group of $\mathbb{C}^n$.

Fundamental definitions on the matrix involved in the complex BSS problem can be done [188, 195].

Definition 18 *Let $\mathbf{A} \in Gl(n; \mathbb{C})$ be an invertible matrix. Then $\mathbf{A}$ is said to be **mixing** if $\mathbf{A}$ has at least two nonzero entries in each row.*

Definition 19 *We say two matrices $\mathbf{A}, \mathbf{W} \in \mathbf{Mat}(\mathbf{m} \times \mathbf{n}; \mathbb{C})$ are **equivalent**, $\mathbf{W} \sim \mathbf{A}$, if $\mathbf{W}$ can be written as $\mathbf{W} = \mathbf{APA}$ with an invertible complex diagonal matrix (scaling matrix) $\mathbf{\Lambda} \in Gl(n; \mathbb{C})$ and an invertible real matrix with unit vectors in each row (permutation matrix) $\mathbf{P} \in Gl(n; \mathbb{R})$. Note that permutation and scaling matrices commute, so $\mathbf{W} = \mathbf{APA} = \mathbf{A\Lambda P}$.*

And similarly

Definition 20 *$\mathbf{A}$ is said to be **scaling-equivalent** to $\mathbf{W}$, $\mathbf{A} \sim_{\mathbf{s}} \mathbf{W}$, if $\mathbf{W} = \mathbf{A\Lambda}$ holds, and $\mathbf{A}$ is **permutation-equivalent** to $\mathbf{W}$, $\mathbf{A} \sim_{\mathbf{p}} \mathbf{W}$, if $\mathbf{W} = \mathbf{AP}$. Therefore, if $\mathbf{A}$ is scaling- or permutation-equivalent to $\mathbf{W}$, it is equivalent to $\mathbf{W}$, but not vice-versa.*

For the case of nonlinear mixing environment additional definitions and theorems are required.

Definition 21 *Let $\mathbf{A} \in Gl(n; \mathbb{C})$ be an invertible matrix, then $\mathbf{A} = (a_{ij})_{i,j=1\ldots n}$ is said to be **absolutely degenerate** if there are two columns $l \neq m$ such that $a_{il}^2 = \lambda a_{im}^2$ for a $\lambda \neq 0$, i.e. the the normalized columns differ only by the signs of the entries.*

Definition 22 *Given a function $\mathbf{f} = \mathbf{f}_R + j\mathbf{f}_I : U \to \mathbb{C}$ assume there exist $a, b, c, d \in \mathbb{R}$ such that at least one is not of absolute value 0 or 1. If $\mathbf{f}_R(a\mathbf{x}_R) = b\mathbf{f}_R(\mathbf{x}_R)$ and $\mathbf{f}_I(c\mathbf{x}_I) = d\mathbf{f}_I(\mathbf{x}_I)$ for all $\mathbf{x} = \mathbf{f}_R + j\mathbf{f}_I \in U$ with $a\mathbf{x}_R, c\mathbf{x}_I \in U$; then $\mathbf{f}$ is said to be (a, b)-**homogeneous** or simply **homogeneous**.*

Let $\mathbf{x} = \mathbf{x}_R + j\mathbf{x}_I$ be a complex random variable, it is possible to define its augmented representation as $\mathbf{x}_{\mathbb{R}} = [\mathbf{x}_R \;\; \mathbf{x}_I]^T$ (see appendix B.2). In this way we can extend the definition 16 to the complex domain.

Definition 23 *We call a complex random vector $\mathbf{X} \to \mathbf{X}_{\mathbb{R}}$ with density $p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}})$* ***bounded****, if its density $p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}})$ is* ***bounded****. Denote $supp\ p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}}) := \{\mathbf{x}|p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}}) \neq 0\}$ the support of $p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}})$ i.e. the closure of the nonzero points of $p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}})$.*
We further call an independent complex random vector $\mathbf{X} \to \mathbf{X}_{\mathbb{R}}$ ***fully bounded****, if $supp\ p_{\mathbf{X}_{\mathbb{R}i}}(\mathbf{X}_{\mathbb{R}i})$ is an interval for all i. So we get $supp\ p_{\mathbf{X}_{\mathbb{R}}}(\mathbf{X}_{\mathbb{R}}) = [a_1, b_1] \times \cdots \times [a_n, b_n]$.*

Remembering the Definition 17 in section 2.7, it is possible to give the following

Definition 24 *A function $\mathbf{f} = \mathbf{f_R}(\mathbf{x_R}) + j\mathbf{f_I}(\mathbf{x_I}) : \mathbb{C}^n \to \mathbb{C}^n$ is called diagonal or component-wise if each component $f_{Ri}(\mathbf{x}_R)$ of $\mathbf{f}_R(\mathbf{x}_R)$ depends only on the variable x_{Ri} and if each component $f_{Ii}(\mathbf{x}_I)$ of $\mathbf{f}_I(\mathbf{x}_I)$ depends only on the variable x_{Ii}.*

In this case we often omit the other variables and write $\mathbf{f}_R(x_{R1}, \ldots, x_{Rn}) = (f_{R1}(x_{R1}), \ldots, f_{Rn}(x_{Rn}))$ or $\mathbf{f}_R = f_{R1} \times \cdots \times f_{Rn}$ and $\mathbf{f}_I(x_{I1}, \ldots, x_{In}) = (f_{I1}(x_{I1}), \ldots, f_{In}(x_{In}))$ or $\mathbf{f}_I = f_{I1} \times \cdots \times f_{In}$.

Consider now the postnonlinear blind source separation model in eq. (4.3). We assume the components f_{Ri} of $\mathbf{f}_R$ and f_{Ii} of $\mathbf{f}_I$ to be injective analytic functions with nonvanishing derivatives. Then also the f_{Ri}^{-1} and f_{Ii}^{-1} are analytic.

Complex postnonlinear BSS is a generalization of linear BSS, so the indeterminacies of complex postnonlinear ICA contain at least the indeterminacies of linear BSS: $\mathbf{A}$ can only be reconstructed up to *scaling* and *permutation*: the scaling ambiguity in complex case is reflected in a *rotation* ambiguity too. Here of course additional indeterminacies come into play because of *translation*: f_{Ri} and f_{Ii} can only be recovered up to a constant. Also, if $\mathbf{\Lambda} \in Gl(n; \mathbb{C})$ is a scaling matrix, then $\mathbf{f}(\mathbf{As}) = (\mathbf{f} \circ \mathbf{\Lambda})((\mathbf{\Lambda^{-1}A})\mathbf{s})$, so $\mathbf{f}$ and $\mathbf{A}$ can interchange scaling factors in each component. Another indeterminacy could occur if $\mathbf{A}$ is not mixing, i.e. at least one observation x_i contains only one source; in this case f_i can obviously not be recovered. For example if $\mathbf{A} = \mathbf{I}$ then $\mathbf{f}(\mathbf{s})$ is already again independent, because independence is invariant under component-wise nonlinear transformation; so $\mathbf{f}$ cannot be found using this method.

If we however assume that $\mathbf{A}$ is mixing and not absolutely degenerate, then we will show for all fully bounded sources $\mathbf{s}$ that except for scaling interchange between $\mathbf{f}$ and $\mathbf{A}$ no more indeterminacies than in the affine complex linear case exist. Note that if $\mathbf{f}$ is only assumed to be continuously differentiable, then additional indeterminacies come into play.

4.3.1 Separability

In this section it is proven separability of postnonlinear BSS; it will be seen how the two conditions from Definition 18 and Definition 21 turn out to be necessary.

Theorem 14 (Separability of bounded complex postnonlinear BSS) *Let* $\mathbf{A}$, $\mathbf{W} \in Gl(n;\mathbb{C})$ *and one of them mixing and not absolutely degenerate,* $\mathbf{h} : \mathbb{C}^n \to \mathbb{C}^n$ *be a diagonal injective analytic function such that* $h_i' \neq 0$ *and let* $\mathbf{s}$ *be a fully bounded independent random vector. If* $\mathbf{W}(\mathbf{h}(\mathbf{As}))$ *is independent, then there exists a scaling* $\mathbf{\Lambda} \in Gl(n;\mathbb{C})$ *and* $\mathbf{v} \in \mathbb{C}^n$ *with* $\mathbf{\Lambda A} \sim \mathbf{W}^{-1}$ *and* $\mathbf{h}(\mathbf{x}) = \mathbf{\Lambda x} + \mathbf{v}$.

So let $\mathbf{f} \circ \mathbf{A}$ be the complex mixing model and $\mathbf{W} \circ \mathbf{g}$ the complex separating model. Putting the two together we get the above complex mixing-separating model with $\mathbf{h} := \mathbf{g} \circ \mathbf{f}$. The theorem shows that if the mixing-separating model preserves independence then it is essentially trivial i.e. $\mathbf{h}$ affine linear and the matrices equivalent (up to scaling and rotation). As usual, the model is assumed to be invertible, hence identifiability and uniqueness of the model follow from the separability.

4.4 Entropic contrasts in the complex environment

The aims of this section is to extend to the complex domain the application of the entropic contrasts, already introduced for the real domain in Section 2.8.

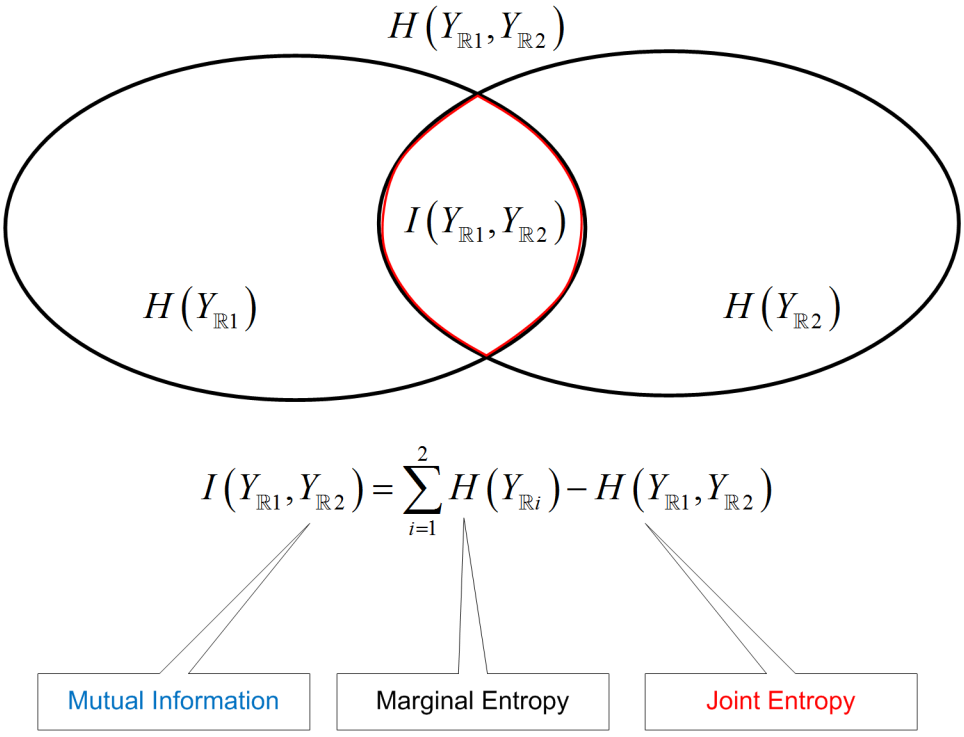

Fig. 4.3: Graphical relation between Mutual Information, Marginal Entropy and Joint Entropy in complex domain

The methodology is the same that in the case of real mixture: in fact the contrasts are real-valued function. In this sense such functions, like *joint entropy* or *mutual information*, are estimated on the augmented representation of the complex vectors $\mathbf{Y}_{\mathbb{R}}$. The relation between Mutual Information and Joint Entropy of a random vector $\mathbf{Y}_{\mathbb{R}}$, can be expressed as (see [55] and Appendix A)

$$I(\mathbf{Y}_{\mathbb{R}}) = \sum_{i=1}^{N} H\left(Y_{\mathbb{R}i}\right) - H(\mathbf{Y}_{\mathbb{R}}) \tag{4.5}$$

A conceptual description of the eq. (4.5) can be described as followed and is showed for $N = 2$ in Figure 4.3.

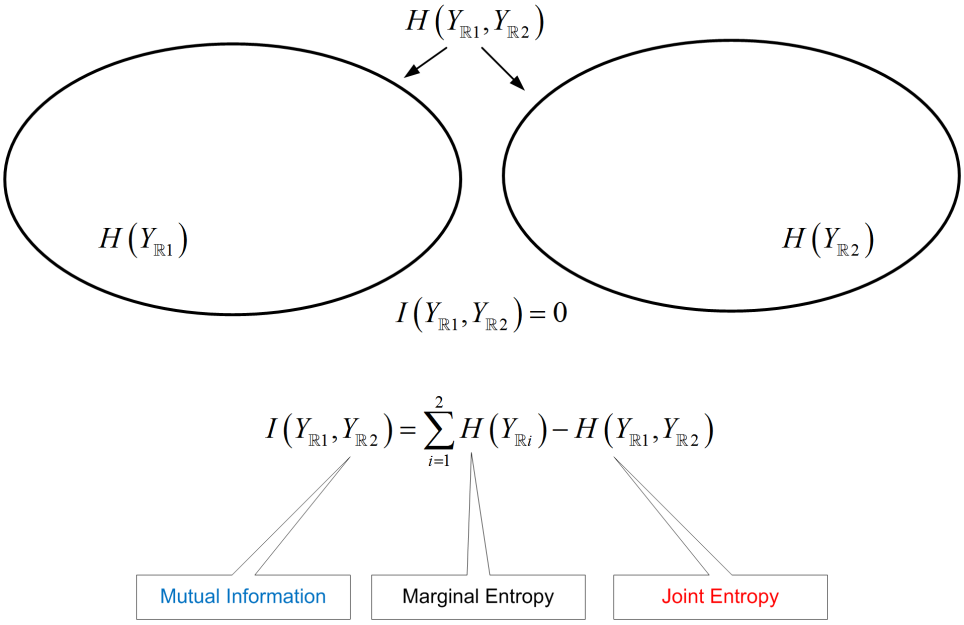

Fig. 4.4: Graphical interpretation of MMI, ME and MNG approaches in complex domain

Because the equation (4.5) is formed by three terms, even in the complex domain one can proceed through three ways to reach the independence of the random variable $Y_{\mathbb{R}i}$. It is possible to minimize the left side $I(\mathbf{Y}_{\mathbb{R}})$ of eq. (4.5) obtaining the *Minimal Mutual Information* or *MMI* approach; it is possible to maximize the joint entropy $H(\mathbf{Y}_{\mathbb{R}})$ in eq. (2.7) obtaining the *Maximum Entropy* or ME approach and it is possible to maximize the term $\sum_{i=1}^{N} H\left(Y_{\mathbb{R}i}\right)$ obtaining the *Maximum Non-Gaussianity* or (*MNG*) approach.

The Figure 4.4 shows that the independence between $Y_{\mathbb{R}1}$ and $Y_{\mathbb{R}2}$ is maximized when the intersection of the two sets is minimized, ideally vanished. This fact is the same that maximizing the union of the two sets or

maximizing the non-covered area of the two single sets. In other words, the independence is guaranteed if it is minimized the mutual information of $\mathbf{Y}_{\mathbb{R}}$ (MMI approach), maximized the joint entropy of $\mathbf{Y}_{\mathbb{R}}$ (ME approach) or maximized the marginal entropy of $Y_{\mathbb{R}1}$ and $Y_{\mathbb{R}2}$ (MNG approach), respectively.

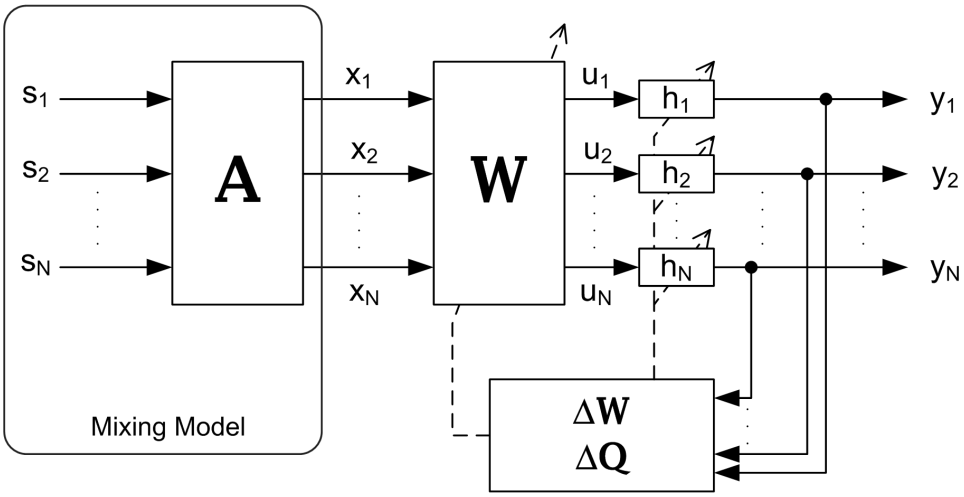

Fig. 4.5: System to perform Blind Source Separation in linear environment

To give an idea of the particular system adopted in the solution of the BSS problem, it is possible to see the Figure 4.5 for the linear environment and the Figure 4.6 for the nonlinear environment, where $\mathbf{W}$ is the de-mixing network while the functions h_i, $i = 1, \ldots, N$ are utilized for evaluating the above functional MMI, ME or MNG.

In particular given the complex network outputs $\mathbf{y} = \mathbf{y}_R + j\mathbf{y}_I$, where $\mathbf{y}_R = [y_{R1}, y_{R2}, \ldots, y_{RN}]^T$ and $\mathbf{y}_I = [y_{I1}, y_{I2}, \ldots, y_{IN}]^T$, it is constructed the augmented vector $\mathbf{y}_{\mathbb{R}} = [\mathbf{y}_R \;\; \mathbf{y}_I]^T$ on which evaluating the MMI, ME or MNG contrasts.

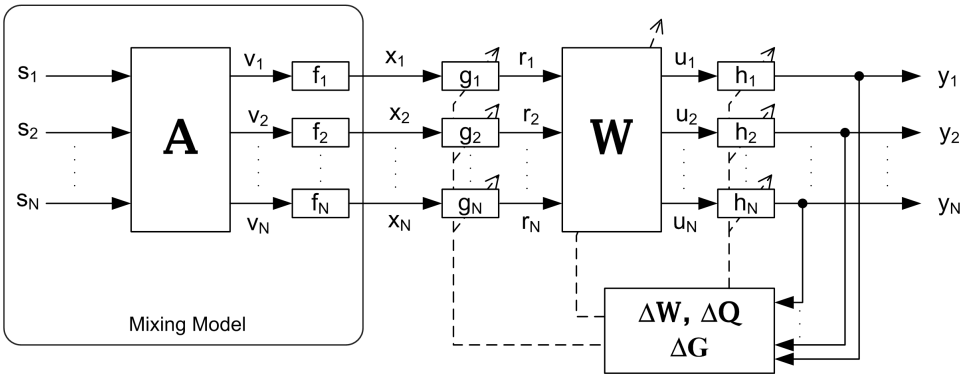

Fig. 4.6: System to perform Blind Source Separation in nonlinear environment

5

The Complex Activation Function

—All cultural products contain a mixture of two elements: conventions and inventions.
J. G. Cawelti

ONE of the main issues in designing complex neural networks is the presence of complex nonlinear functions involved in the learning processing [30, 146], i.e. complex activation functions or distorting functions (for the nonlinear mixing environment): to ensure the universal approximation network capabilities, the activation functions should be bounded and differentiable.

5.1 The main challenge

Let $h(z)$ be a complex nonlinear *activation function (AF)*, where $z = z_R + jz_I \in \mathbb{C}$, z_R and z_I are the real and imaginary parts of the complex variable z.

The main challenge is the dichotomy between boundedness and analyticity in the complex domain [103], as stated by the *Liouville's theorem* (see 24): *complex functions, bounded on the whole complex plane, are either constant or not analytic*. Thus this kind of complex nonlinear functions are not suitable as activation functions of neural networks.

Georgiou and Koutsougeras in [69] defined five properties which should be satisfied by complex nonlinear functions in neural network applications:

1. $h(z) = h(z_R, z_I) = h^R(z_R, z_I) + jh^I(z_R, z_I)$ is nonlinear in z_R and z_I;
2. $h(z)$ is bounded: $|h(z)| \leq c < \infty$;
3. $h^R_{z_R}, h^R_{z_I}, h^I_{z_R}, h^I_{z_I}$ exist and are bounded;

4. $h(z)$ is not entire[1];

5. $h^R_{z_R} h^I_{z_I} \neq h^R_{z_I} h^I_{z_R}$,

where $h^R(z_R, z_I)$ and $h^I(z_R, z_I)$ are known as the real part function and imaginary part function of the complex function $h(z)$ respectively, while $h^R_{z_R} = \frac{\partial h^R}{\partial z_R}$, $h^R_{z_I} = \frac{\partial h^R}{\partial z_I}$, $h^I_{z_R} = \frac{\partial h^I}{\partial z_R}$, $h^I_{z_I} = \frac{\partial h^I}{\partial z_I}$.

It should be noted that the properties in 1-5 require the boundedness of the nonlinear function and its derivatives even when the function is defined in a local domain [26]. By the Liouville's theorem the cost for this restriction is that the function is not analytic.

The boundedness of the AF is essential to prove the universal approximation of a complex feed-forward neural network [104, 102, 100]. These works have shown that, for the multilayer perceptron (MLP), a complex counterpart of the universal approximation theorem can be realized with activation functions that are entire (analytic for all values of z) but bounded only almost everywhere. This is an extension of the real-valued result in Cybenko in [56] to the complex case.

In this context Kim and Adali proposed in [104, 103, 101] the use of the so-called *elementary transcendental functions (ETF)*. They classified the ETFs into two categories of unbounded functions, depending on which kind of singularities[2] they possess. The following functions are noted to provide the nonlinear decorrelation required for ICA when used for the nonlinear activation function $h(z)$:

- Circular functions: $tan(z)$, $sin(z)$ and $cot(z)$;
- Inverse circular functions: $tan^{-1}(z)$, $sin^{-1}(z)$ and $cos^{-1}(z)$;
- Hyperbolic functions: $tanh(z)$, $sinh(z)$ and $coth(z)$;
- Inverse hyperbolic functions: $tanh^{-1}(z)$, $sinh^{-1}(z)$ and $cosh^{-1}(z)$.

As expected the trigonometric and the corresponding hyperbolic functions behave very similarly.

These transcendental functions are entire (analytic) and bounded almost everywhere, i.e. they are unbounded only on a set of points having zero measure. If used as AFs in neural networks they assure convergence almost everywhere.

[1] A function $h(z)$ is said analytic in z_0 if its derivative exists throughout some neighbourhoods of z_0. If $h(z)$ is analytic in all points $z \in \mathbb{C}$, it is called *entire*.

[2] A *singularity* is a point in which a function is not analytic and thus not differentiable: if $\lim_{z \to z_0} h(z) \to \infty$ but the function is analytic in a deleted *neighbourhood* of z_0 (that is a *pole*), the singularity is said to be *isolated*; if $\lim_{z \to z_0} h(z)$ exists it is isolated but *removable*; if none of these cases are met, the function has an isolated *essential* singularity.

Moreover, in signal processing applications where the domain of interest is a bounded neighborhood of the unit circle these singular points scarcely pose a problem.

Recently Adali et al. in [2] have used these above ETFs as complex activation functions in BSS problems. In particular, the functions: $tan^{-1}(z)$, $sin^{-1}(z)$, $cos^{-1}(z)$ and $tan(z)$, and their hyperbolic counterparts performed consistently well over a wide range of input and mixtures, while the functions: $sin(z)$, $cos(z)$, $cot^{-1}(z)$, $sinh(z)$, $cosh(z)$ and $coth^{-1}(z)$ exhibited unstable behavior when used for ICA with the algorithm proposed by Bell and Sejnowski in [15] and described later.

A compromised approach to process real and imaginary components of complex signal jointly can be found in [69, 81]. These authors proposed joint-nonlinear complex activation functions that process the real and imaginary components as shown in the following equations:

$$h(z) = \frac{z}{c + \frac{|z|}{r}} \tag{5.1}$$

$$h(\alpha \cdot \exp[j\beta]) = \tanh\left(\alpha/m\right) \exp(j\beta) \tag{5.2}$$

where, c and r are real positive constants, and m is a constant that is inversely related to the gradient of the absolute function $|h|$ along the radius direction around the origin of the complex coordinate for $z = \alpha \cdot exp(j\beta)$. However, these functions are still not analytic and also preserve the phase. The inability to provide accurate nonlinear phase response poses a significant disadvantage for these functions in signal processing applications.

5.2 The *splitting* solution

According to the properties in 1-5 listed above, in order to overcome the dichotomy between boundedness and analyticity, complex nonlinear *splitting functions* have been introduced. In this approach real and imaginary parts are processed separately by real-valued nonlinear functions [19, 174, 200]. The splitting function

$$h(z) = h(z_R, z_I) = h^R(z_R) + jh^I(z_I) \tag{5.3}$$

avoids the problem of unboundedness of complex nonlinearities, as stated above, but it cannot be analytic (see Figure 5.1).

Even though bounded, the complex activation function defined in this way is not analytic and the *back-propagation* phase during the learning process also takes split paths through disjoint real-valued gradients. As would be expected, such a scheme will not be efficient when learning nonlinear mappings of complex input/output pairs [104, 102, 100].

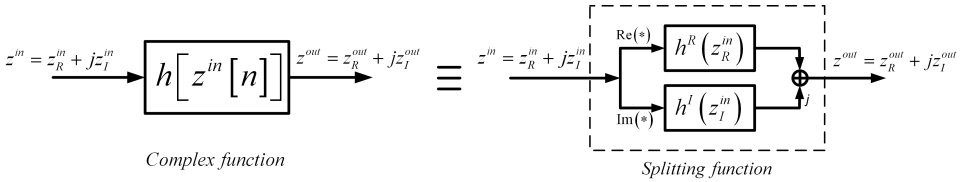

Fig. 5.1: Splitting function realization of a complex activation function (AF)

The splitting model of a nonlinear complex valued function is not realistic because usually the real and imaginary part are correlated. According to this issue it is useful to perform a more realistic model of the nonlinear functions. In this way, Vitagliano et al. in [214] proposed a complex neural network based on a couple of bi-dimensional functions (Figure 5.2) called generalized splitting function:

$$h(z) = h(z_R, z_I) = h^R(z_R, z_I) + jh^I(z_R, z_I). \tag{5.4}$$

We consider each part $h^R(z_R, z_I)$ and $h^I(z_R, z_I)$ as two functions of two variables: one plays the role of the real part function and one of the imaginary part function of the complex activation function. With regard to the *"desired properties"* stated for the fully complex AFs we can note that the generalized splitting function:

- it is a nonlinear function with respect to the coordinates; thus $h(z)$ is a nonlinear function with respect to z_R and z_I;
- it has not singularities and it is bounded for each $z = z_R + jz_I$;
- the partial derivatives $h^R_{z_R}, h^R_{z_I}, h^I_{z_R}, h^I_{z_I}$ are continuous and bounded;
- the condition $h^R_{z_R} h^I_{z_I} \neq h^R_{z_I} h^I_{z_R}$ is verified.

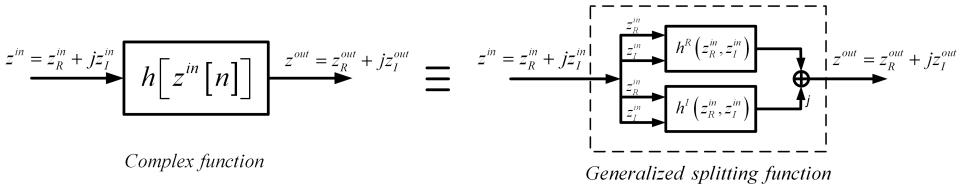

Fig. 5.2: Generalized splitting function realization of a complex activation function (AF)

In this way $h(z)$ is bounded but it is not analytic. The *Cauchy-Riemann conditions* ($h^R_{z_R} = h^I_{Z_I}$, $h^I_{z_R} = -h^R_{Z_I}$) are not satisfied by the complex function in eq. (5.4) itself, but can be imposed by an algorithm constraint during the learning process: $h^R_{z_R} = h^I_{z_I} = \left(h^R_{z_R} + h^I_{z_I}\right)/2$ and $h^R_{z_I} = -h^I_{z_R} =$

$\left(h^I_{z_R} + h^R_{z_I}\right)/2$. Note that the Cauchy-Riemann conditions are equivalent to the fifth property in 1-5.

Another approach for adapting the nonlinearity to the source distribution is introduced in [3]. It is based on the idea that the simple substitutions $z_R = (z + z^*)/2$ and $z_I = (z - z^*)/2j$ (where $(\bullet)^*$ denotes the complex conjugate) allow us to write a given pdf that is $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$ in terms of a function $\mathbb{C} \times \mathbb{C} \to \mathbb{R}$.

This approach is based on the Brandwood's result [22]. Let we define $h : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ as a function of a complex variable z and its conjugate z^*. If treating z (respectively z^*) as a constant, h is analytic on z^* (respectively z), then we say that h satisfies the *Brandwood's analyticity condition* (*BAC*). Because the main interest is in functions h that are cost functions, it is possible to consider the more special case of $h : \mathbb{C} \times \mathbb{C} \to \mathbb{R}$. Then the main result of [22] for these class of functions can be expressed as [3]:

Theorem 15 *Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a function of real variables z_R and z_I such that $h(z, z^*) = f(z_R, z_I)$, where $z = z_R + jz_I$ and that h satisfies the BAC. Then, the partial derivative $\partial h/\partial z$ (treating z^* as a constant in h) gives the same result as $(\partial f/\partial z_R - j\partial f/\partial z_I)/2$ on substituting for z. Similarly, $\partial h/\partial z^* = (\partial f/\partial z_R + j\partial f/\partial z_I)/2$.*

In [3] the authors applied this result on the maximization of the complex Likelihood for BSS application.

6

Spline Functions

—Models are to be used, not believed.
H. Teil

A PROBLEM in choosing complex AF is the matching between the shape of the AF and the cumulative density function (cdf) of the unknown sources [218]. In fact the separation can be well performed in the case that the AF coincides with the source cdf, as explained hereinafter (see eq. (7.11)). The idea is to adopt a flexible solution [42]: the shape of the AF is adaptively changed from data by the use of flexible functions, performing the so-called Flexible ICA. This solution allows the separation of signal with a no pre-defined cumulative density function (cdf) and moreover increases the quality of the separation.

The problem addressed in this section is to find a nonlinear adaptive function (or a curve), suitable for implementing an activation function, that: 1) satisfies the boundedness constraint defined for activation functions [41, 56, 85, 182, 181], 2) is able to retain the universal approximation property and 3) is flexible enough to modify its shape by adapting a small number of parameters.

6.1 Introduction

In the last years an increasingly interest in adaptive activation functions has arisen. The simplest solution consists in involving a parametric gain and slope of a sigmoid AF in the learning process. A different approach is based on the use of polynomial functions which allows reducing the size of the network and the connection complexity [201]. The digital implementation of this kind of activation function through a look-up-table (LUT) keeps the complexity under control [147] and is easy to realize. The LUT values

can be seen as the curve sample points and one can think to iteratively adapt them in order to change the function shape. Only after a certain number of adaptation steps the shape of the activation function can reflect the information represented by data.

However the direct use of a LUT activation function can lead to a huge number of free parameters, so it is more desirable using a suitable interpolation or approximation scheme. The choice of these schemes is not an obvious one: a wrong choice of the interpolation scheme can lead to problems in the development of the learning algorithm [19, 74, 201, 175, 177].

A good interpolation scheme should guarantee a continuous first derivate and the capability to locally adapt the curve: such properties are exhibited by the so-called *piecewise polynomial spline interpolation scheme*. For BSS we prefer an interpolation scheme, due to its local characteristics, which avoids the oscillatory behaviour of the global adaptation of the approximation scheme. There are few splines that interpolate their control points. One that also has a very low computational overhead is the so-called *Catmull-Rom cubic spline (CR)* [39, 207, 201, 200]. There is a regularization property common to most of the polynomial spline basis sets called *variation diminishing property* [54, 169, 124], which ensures the absence of unwanted oscillations of the curve between two consecutive control points. So we can have an exact representation of linear segments.

6.1.1 Spline in regularization theory

The reconstruction of a curve from the knowledge of a finite set, T_N, of samples is a typical ill-posed problem; in general, there are many possible functions with good approximation capabilities on the pairs (x_i, t_i) and not all of them can be considered in the same way. In fact, the presence of noise in the measurements induces, in the space X, high-frequency components that are only a disturbance. At the same time, in those regions of X in which we donŠt have many samples, the neural network should give a smooth approximation of the few available data, avoiding unjustified oscillations.

Regularization theory offers a way to choose a compromise between data fitting and smoothness, through a regularizing term added to the classical squared error and weighted by a constant:

$$H(f) = \sum_{i=1}^{N} \left[t_i - f(\mathbf{x}_t)\right]^2 + \lambda \|Pf\|^2 \tag{6.1}$$

where $H(f)$ represents the functional to be minimized. The stabilizer P is the differential operator determining the kind of smoothness and the shape of the approximator, while $\|\bullet\|$ is a suitable norm. It is well known that the

minimization of eq. (6.1) leads to

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i G(\mathbf{x} - \mathbf{x}_i) \tag{6.2}$$

G is the Green's function corresponding to the operator P and c_i are coefficients determined by solving the $N \times N$ linear system $(\mathbf{G} + \lambda \mathbf{I})\mathbf{c} = \mathbf{t}$, where $\mathbf{G}$ is the Green's matrix, $\mathbf{c}$ is the column vector of the coefficients and $\mathbf{t}$ is the column vector of the targets t_i.

In particular, we are interested in the one-dimensional stabilizer

$$\|Pf\|^2 = \int_R \left[\frac{d^2 f(x)}{dx^2}\right]^2 dx \tag{6.3}$$

which corresponds to the kernel $\mathbf{G}(x) = |x|^3$. For the multidimensional case, in [70] it is shown that we can use the same stabilizer, just decomposing the function f in the sum of n functions, each in charge of one component of the vector $\mathbf{x}$

$$f(\mathbf{x}) = \sum_{j=1}^{n} f_j(x_j) \tag{6.4}$$

Then, the overall kernel is

$$\bar{G}(x) = \sum_{j=1}^{n} \mu_j G(x_j) = \sum_{j=1}^{n} \mu_j |x_j|^3 \tag{6.5}$$

where μ_j, $j = 1, \ldots, n$, are constants. The final aspect of the approximating function is

$$f(\mathbf{x}) = \sum_{i=1}^{N} c_i \sum_{j=1}^{n} \mu_j G(x_j - x_{ij}) \tag{6.6}$$

the symbol x_{ij} indicates the j-th component of the i-th input x_i in T_N. An important extension of the previous function involves a change in the system of coordinates for the space X; as reported in [70], the choice of a proper "point of view" can be important when representing a multivariate function as the sum of a number of functions equal to the dimension of the input space. Calling w_j, $j = 1, \ldots, n$, the vectors which determine the axis of the new system and α_{ij} the new centers in such a system, we can write, inverting the order of summation of eq. (6.6),

$$f(\mathbf{x}) = \sum_{j=1}^{n} \mu_j \sum_{i=1}^{N} c_i G(\mathbf{w}_j \mathbf{x} - \alpha_{ij}) \tag{6.7}$$

that is the starting point of our considerations.

First of all, we need to evaluate the fixed parameters m_j; with no a priori assumption we may use data-driven procedures, i.e., cross-validation, which can be computationally expensive when applied to large nonlinear models like most neural networks. Then a $N \times N$ linear system must be solved in order to find the coefficients c_i, $i = 1, \ldots, N$; in real applications, the number N can be large and the problems related to the inversion of the matrix $(\mathbf{G} + \lambda \mathbf{I})$ can become quite hard. The large value of N is an obstacle for hardware implementation, too, because we should use nN kernels (N kernels for each of the n directions), which means nN neurons, a lot of connections and, in the case of VLSI implementation, a large silicon area. Of course, it is possible to reduce the number of centers from N to N' ($N' < N$) using a sub-optimal solution of the minimization of eq. (6.1), but here we will take a different approach.

Our idea consists in realizing a neuron with a more complex activation function than the sigmoid, able to reproduce the shape of a whole cubic spline along the directions specified by $\mathbf{w}_j$, $j = 1, \ldots, n$.

$$\phi\left(\mathbf{w}_j \mathbf{x}\right) = \sum_{i=1}^{N} c_i \left|\mathbf{w}_j \mathbf{x} - \alpha_{ij}\right|^3, \; j = 1, \ldots, n \tag{6.8}$$

Then $f(x)$ can be written as

$$f\left(\mathbf{x}\right) = \sum_{j=1}^{n} \mu_j \phi_j \left(\mathbf{w}_j \mathbf{x}\right) \tag{6.9}$$

Now μ_j, and the components of $\mathbf{w}_j$, for all the indexes j, can be found by backpropagation, thus solving the problem of the optimal set of the parameters μ_j and of the ideal system of coordinates (although we can get trapped in local minima). The open question is about φ_j. Once again, the exact implementation of eq. (6.8) would require the knowledge of all the coefficients c_i, so we choose a different solution, that is using a cubic spline of simpler structure. Its main characteristics are the adaptation of its shape through some control points and a suitable degree of smoothness. Notice that in our implementation a bias parameter w_{j0} similar to the one used in sigmoidal neurons has been introduced and so we will deal with a function $\varphi_j(\mathbf{w_j x} + \mathbf{w_{j0}})$. The last point to discuss is the approximation capability of the function in eq. (6.8): of course, it will behave well on targets which are likely to have an additive structure, but, in general, a number of hidden units equal to the dimension of the input space is not enough to obtain universal approximation of continuous functions on a compact set. However, we can extend the idea of a neuron with a cubic and smooth activation function to architectures involving a larger number of hidden units (or even more than one hidden layer), though they cannot be directly derived from regularization theory.

6.2 Matrix formulation of Spline function

A planar spline curve is a two-dimensional vector, whose components are piecewise polynomial univariate functions of the same degree: its mathematical formulation ensures both its continuity and the existence of its derivatives, along the curve and in correspondence to the joining points between the various curve spans [74]. Given a LUT as defined above, a general spline expression for that curve would be

$$F(\nu) = \left[\begin{array}{cc} F_u(\nu) & F_y(\nu) \end{array} \right]^T = \overset{N-3}{\underset{i=0}{\mathbf{C}}} F_i(\nu) \tag{6.10}$$

where **C** is the concatenation operator and $F_i(u)$ of i-th curve span (or patch). The indixes **C** of the operator in (6.10) are valid only for cubic polynomials: the choice of using cubic polynomials was made because of the trade-off between the requested properties and the computational complexity. Figure 6.1 shows the superposition of $N-3$ cubic spline spans.

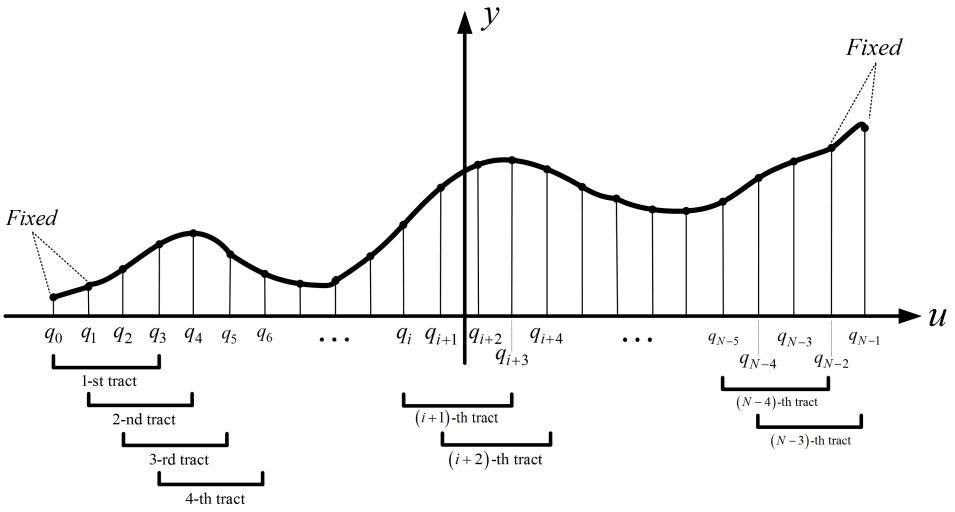

Fig. 6.1: The AF as a superposition of $N-3$ cubic spline spans

The parameter ν has the property of being *local* and its domain is $0 \leq \nu \leq 1$ for every curve span [74]. Hence, there must be a unique mapping that allows us to calculate the *local* parameter ν, as well as the proper curve span i, from the abscissa *global* parameter. In this way, we can represent any point lying on the spline curve $F(\nu)$ as a point belonging to the single $F_i(\nu)$ curve span. It follows (see [13]) that the i-th curve span can be described as follows:

$$F_i(\nu) = \left[\begin{array}{cc} F_{ui}(\nu) & F_{yi}(\nu) \end{array} \right]^T = \sum_{j=0}^{3} Q_{i+j} C_j(\nu) \tag{6.11}$$

where $C_j(\nu)$ are the spline polynomials. As in (6.10), each coordinate is described by a univariate function, namely a cubic polynomial of the variable ν.

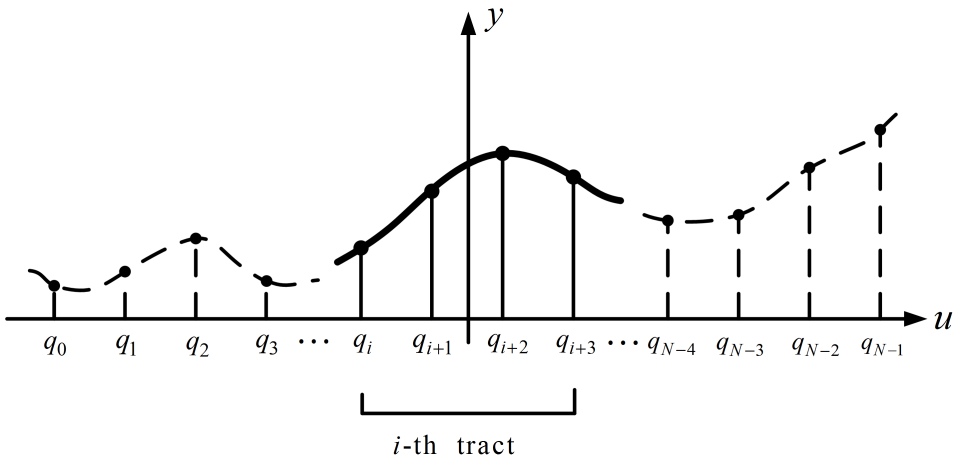

Fig. 6.2: i-th tact of the spline function

Equation (6.11) resolves the link between the curve and the sample points Q_i which are called *control points* in the literature, as they control the shape of the curve. The cubic polynomial functions $C_j(\nu)$, called *spline basis functions* or *blending functions*, characterize the way the curve moves along the path made up by the control points. The curve can interpolate or just approximate its control points, depending on which blending function set we rely on (see for example [13]).

Let u be the input of the activation function of a neuron; we have to make the dependence between u and $F_i(\nu)$ explicit. There is a direct link between u and one of the two components (a cubic polynomial function) [74], namely

$$u = F_{ui}(\nu) \tag{6.12}$$

Equation (6.12) plays a key role in finding a suitable activation function architecture, as it is the main bottleneck of the overall structure: in fact, it is the only link between the input u and nonlinear neuron output $F_{yi}(\nu)$, which is given in term of the $i \in [0, N]$ value (which represents the LUT index) and the $\nu \in [0, 1]$ value (which represents the offset of the i-th curve span). Let y be the neuron output, we have to introduce a two-step procedure to calculate y of a single neuron, given the input u [74]:

1. calculate ν and i from s by inverting (6.12);
2. substitute these values of ν and i in $y = F_{yi}(\nu)$.

Now there are several splines that interpolate their control points; one that also has a very low computational overhead is the so called *Catmull-Rom*

(CR) cubic spline basis [39], which is described by the following polynomials [with reference to (6.11)]:

$$\begin{aligned} C_0(\nu) &= \frac{1}{2}\left(-\nu^3 + 2\nu^2 - \nu\right) \\ C_1(\nu) &= \frac{1}{2}\left(3\nu^3 - 5\nu^2 + 2\right) \\ C_2(\nu) &= \frac{1}{2}\left(-3\nu^3 + 4\nu^2 + \nu\right) \\ C_3(\nu) &= \frac{1}{2}\left(\nu^3 - \nu^2\right). \end{aligned} \tag{6.13}$$

A quick inspection of (6.13) shows that all the multiplications, except for the powers of the parameter ν, are by integer coefficients and that they should be easily implemented in hardware (just one or two shifts or sum-and-shift operations). This characteristics of the CR spline is important, as it simplifies the structure of the nonlinear block. A common expression of (6.11), is the following matrix notation:

$$F_i(\nu) = \begin{bmatrix} \nu^3 & \nu^2 & \nu & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_i \\ Q_{i+1} \\ Q_{i+2} \\ Q_{i+3} \end{bmatrix} \tag{6.14}$$

which explicits the actual array structure of a curve span: a *parameter vector*, a *basis matrix* and a *control point vector*, combined using row by column multiplications.

The general result of (6.14) can be expressed as a pair of cubic polynomials (where a, b, c and d are appropriate constants)

$$\begin{aligned} F_{ui}(\nu) &= a_{xi}\nu^3 + b_{xi}\nu^2 + c_{xi}\nu + d_{xi} \\ F_{yi}(\nu) &= a_{yi}\nu^3 + b_{yi}\nu^2 + c_{yi}\nu + d_{yi} \end{aligned} \tag{6.15}$$

In (6.12), the inversion of the first polynomial in (6.15) would be needed to get the parameters ν and i, which is in turn necessary to generate the neuron's output. We could exploit the formulas for the solution of third-order equations (algebraically or iteratively), but a serious overhead in the calculations would be introduced. Moreover, this approach gives no valid answer to the ordering problem that affects the abscissa control points. There is, however, the regularization property called *variation diminishing property* [54, 169, 124], which ensures the absence of unwanted oscillations of the curve between two consecutive control points, as well as the exact representation of linear segments. This second statement is particularly important, as it suggests a simple possible solution to the inversion problem (6.12): if we uniformly sample the abscissas along the x-axis, then the cubic polynomial $F_{ui}(\nu)$ becomes a *first degree* polynomial. This approach ensures

both a fast computation of the local parameters ν and i, and the monotone ordering of the abscissas, in the case when they are kept fixed.

In this case, let us consider a fixed sample step $\Delta u = q_{u,i+1} - q_{u,i}$: after substituting the proper values of the abscissa control points in (6.14), the function $F_{ui}(\nu)$ has the following form:

$$F_{ui}(\nu) = \begin{bmatrix} \nu^3 & \nu^2 & \nu & 1 \end{bmatrix} \frac{1}{2} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_{u,i} \\ q_{u,i} + \Delta u \\ q_{u,i} + 2\Delta u \\ q_{u,i} + 3\Delta u \end{bmatrix} \tag{6.16}$$

The solution to this equation (6.16) is a linear mapping

$$F_{ui}(\nu) = \nu \Delta u + q_{u,i} + \Delta u = \nu \Delta u + q_{u,i+1} \tag{6.17}$$

in which all the parameters are known. Expression (6.17) is very fast to compute, and it leads to a negligible computational overhead. Moreover, it is not necessary to store the $q_{u,i}$ in the table, as these values can now be determined algorithmically (more on this later).

The (6.15) can now be written as

$$\begin{aligned} F_{ui}(\nu) &= u = c_{xi}\nu + d_{xi} \\ F_{yi}(\nu) &= y = a_{yi}\nu^3 + b_{yi}\nu^2 + c_{yi}\nu + d_{yi} \end{aligned} \tag{6.18}$$

Operating the inversion $\nu = {F_{xi}}^{-1}(u)$, the output y can be computed as a cubic polynomial which expresses a direct relation between the input u and the output y of the CR-based nonlinear block

$$\begin{aligned} y = F_{yi}(u) = \frac{a_{yi}}{c_{xi}^3}u^3 + \left(\frac{b_{yi}}{c_{xi}^2} - \frac{3a_{yi}d_{xi}}{c_{xi}^3}\right)u^2 + \\ + \left(\frac{3a_{yi}d_{xi}^2}{c_{xi}^3} - \frac{2b_{yi}d_{xi}}{c_{xi}^2} + \frac{c_{yi}}{c_{xi}}\right)u + \\ + \left(\frac{b_{yi}d_{xi}^2}{c_{xi}^2} - \frac{a_{yi}d_{xi}^3}{c_{xi}^3} - \frac{c_{yi}d_{xi}}{c_{xi}} + d_{yi}\right) \end{aligned} \tag{6.19}$$

The expression (6.19), although simple to understand, cannot be efficiently calculated in this form: it is for the sake of computational efficiency that we will use (6.14) for the actual computation of $F_{yi}(\nu)$.

The result found in (6.17) is then employed to improve the efficiency of the complete structure. Therefore, we constrain the control point abscissas to be equidistant and, most important, not adaptable. Moreover, always for the sake of efficiency, another constraint is imposed on the control points, forcing the sampling interval to be centered on the x-axis origin. It is then possible to represent the abscissa of each point of the activation function using two parameters (the span index i and the local parameter ν), without

storing the control points abscissas. All we need to know is how many control points the curve has, and the sampling step Δu.

In this sense it is possible to represent the abscissa of each point of the activation function using two parameters: the span index i and the local parameter ν. The i index is used to address the local control points, while the fractional part ν is passed as normalized input to cubic spline function.

A spline approximating its control points is the so-called B-Spline [58, 151, 202]. A comparison between these two schemes is shown in Figure 6.3 which shows graphically the difference between Catmull-Rom Spline and B-Spline for the mono-dimensional spline function; these expressions differ only in the entries of the matrix $\mathbf{M}$ in eq. (6.22).

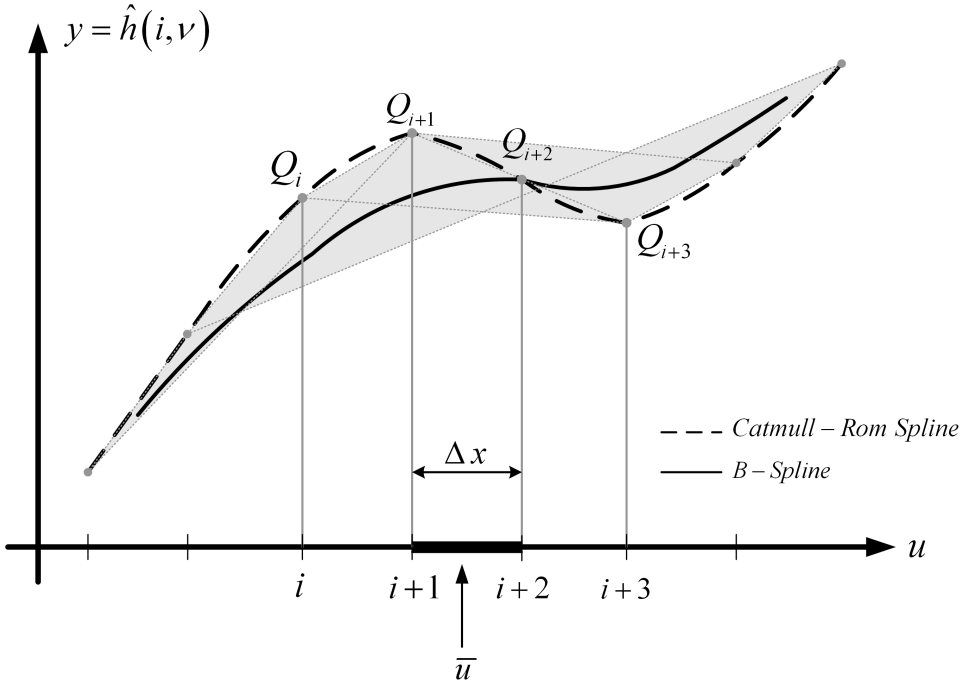

Fig. 6.3: Comparison of C-R spline and B-spline

A common expression of these spline activation functions derived from (6.14), is the following matrix notation:

$$y = \hat{h}\,(i,\nu) = \mathbf{T}_\nu \cdot \mathbf{M} \cdot \mathbf{Q}_i \tag{6.20}$$

which makes explicit the actual parameter vector, a basis matrix and a control points vector, combined using row by column multiplications, where

$$\mathbf{T}_\nu = \begin{bmatrix} \nu^3 & \nu^2 & \nu & 1 \end{bmatrix}, \qquad \mathbf{Q}_i = \begin{bmatrix} Q_i \\ Q_{i+1} \\ Q_{i+2} \\ Q_{i+3} \end{bmatrix} \tag{6.21}$$

where $Q_i = \begin{bmatrix} q_{u,i} & q_{y,i} \end{bmatrix}^T$, $q_{u,i}$ and $q_{y,i}$ are the coordinates of the i-th control point Q_i (contained in the two-dimensional vector y) and $0 \leq \nu < 1$ is the local abscissa. If the abscissas along the u-axis are uniformly sampled, $\Delta u = q_{u,i+1} - q_{u,i} = const$ for every i, then the cubic polynomial becomes a first degree polynomial and the vector y becomes a one-dimensional vector containing only the y-axis of the curve. The matrix $\mathbf{M}$ determines which kind of spline basis is used and has the following expression for the *Catmull-Rom spline* and *B-spline*:

$$\mathbf{M}^{CR} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}, \quad \mathbf{M}^{B} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{6.22}$$

The entire approximation is represented through the concatenation of local cubic spline functions each controlled by 4 control points defined by the two local parameters: i and ν. Given an input value $\bar{u}$ we can calculate these parameters using two internal dummy variables ζ and $\bar{\zeta}$ as follows:

$$\begin{aligned} &\bar{\zeta} = \tfrac{\bar{u}}{\Delta u} + \tfrac{N-2}{2} \\ &\zeta = \begin{cases} 1 & \bar{\zeta} < 1 \\ \bar{\zeta} & 1 \leq \bar{\zeta} \leq N-3 \\ N-3 & \bar{\zeta} > N-3 \end{cases} \\ &i = \lfloor \zeta \rfloor \\ &\nu = \zeta - i \end{aligned} \tag{6.23}$$

where $\lfloor \bullet \rfloor$ is the floor operator that returns the highest integer less than or equal to its input and N is the number of spline control points. In this sense the spline neuron can be considered composed by two blocks: the first one computes the two local parameters i and ν while the second one computes the output neuron y by the local parameters, as shown in Figure 6.4

An additional and important constraint is to force the activation function to be a limiting function, imposing to be constant for $u \to \pm\infty$, while maintaining the ability to modify its shape inside these constant values. We can fix the first two and the last two control points, as shown in Figure 6.1. So for each input $\bar{u}$, we can adapt two points on the left and two on the right, while all the other control points are fixed.

We can generalize the theory discussed above to realize these functions as hyper-surface interpolation of some control points using higher order interpolants ([175, 177, 178]. In particular piecewise of cubic spline are here employed in order to render the hyper-surface continuous in its partial derivatives. The entire approximation is represented through the concatenation of local functions each centered and controlled by $4^2 = 16$ control points, which lie on a regular $2D$ grid in $\mathbb{R}^2$, defined over the region $0 \leq \nu_R, \nu_I < 1$,

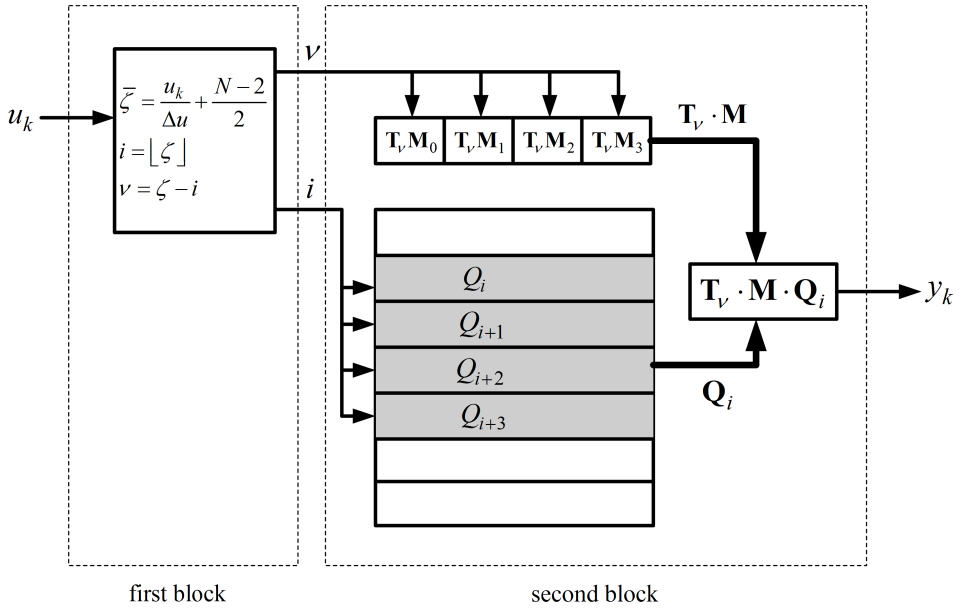

Fig. 6.4: The spline neuron

and in matrix formulation it is expressed as follows:

$$y = \hat{h}(i_R, i_I; \nu_R, \nu_I) = \mathbf{T}_{\nu I} \cdot \mathbf{M} \cdot (\mathbf{T}_{\nu R} \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]}^{(i_R,i_I)})^T \tag{6.24}$$

where $\mathbf{T}_{\nu R} = \begin{bmatrix} \nu_R^3 & \nu_R^2 & \nu_R & 1 \end{bmatrix}$, $\mathbf{T}_{\nu I} = \begin{bmatrix} \nu_I^3 & \nu_I^2 & \nu_I & 1 \end{bmatrix}$, $\mathbf{M}$ is the same defined in (6.22) and $\mathbf{Q}_{[2]}^{(i_R,i_I)}$ is a structure collecting the local control points:

$$\mathbf{Q}_{[2]}^{(i_R,i_I)} = \begin{bmatrix} Q^{(i_R-1,i_I-1)} & Q^{(i_R-1,i_I)} & Q^{(i_R-1,i_I+1)} & Q^{(i_R-1,i_I+2)} \\ Q^{(i_R,i_I-1)} & Q^{(i_R,i_I)} & Q^{(i_R,i_I+1)} & Q^{(i_R,i_I+2)} \\ Q^{(i_R+1,i_I-1)} & Q^{(i_R+1,i_I)} & Q^{(i_R+1,i_I+1)} & Q^{(i_R+1,i_I+2)} \\ Q^{(i_R+2,i_I-1)} & Q^{(i_R+2,i_I)} & Q^{(i_R+2,i_I+1)} & Q^{(i_R+2,i_I+2)} \end{bmatrix}$$

where $Q^{(i_R,i_I)} = \begin{bmatrix} q_{u_R,i} & q_{u_I,i} & q_{y,i} \end{bmatrix}^T$, $q_{u_R,i}$, $q_{u_I,i}$ are the abscissas along the u_R-axis and the u_I-axis and $q_{y,i}$ is the ordinate of the control point $Q^{(i_R,i_I)}$. If the abscissas along the u_R-axis and the u_I-axis are uniformly sampled, the structure $\mathbf{Q}_{[2]}^{(i_R,i_I)}$ becomes a simple 4×4 matrix. An example of cubic $2D$ spline is showed in the following Figure 6.5.

6.2.1 Different spline basis

The spline curve can interpolate or just approximate its control points, depending on which blending function set we rely on [13]. In this sense there are different spline basis matrix $\mathbf{M}$ except those in eq. (6.22).

Here a non comprehensive list of spline basis is related.

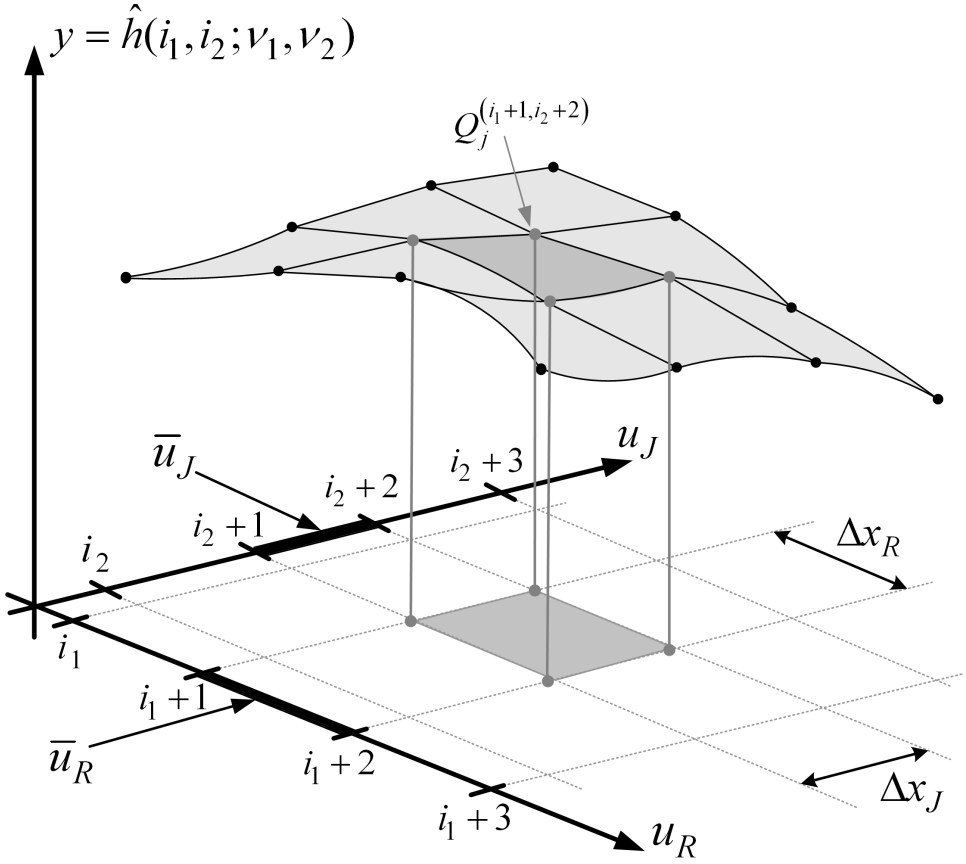

Fig. 6.5: An example of 2D C-R spline

1. **Catmull-Rom** spline basis

$$\mathbf{M}^{CR} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \tag{6.25}$$

2. **B-spline** basis

$$\mathbf{M}^{B} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \tag{6.26}$$

3. **Bezier** spline basis

$$\mathbf{M}^{Bez} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.27}$$

4. **Hermite** spline basis

$$\mathbf{M}^H = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.28}$$

5. **Beta** spline basis, which depends on two parameters β_1 and β_2

$$\mathbf{M}^\beta = \frac{1}{\Delta} \begin{bmatrix} -2\beta_1^3 & 2\beta_2 + 2\beta_1\left(\beta_1^2 + \beta_1 + 1\right) & -2\left(\beta_2\beta_1^2 + \beta_1 + 1\right) & 2 \\ 6\beta_1^3 & -3\beta_2 - 6\beta_1^2\left(\beta_1 + 1\right) & 6\beta_1^2 + 3\beta_2 & 0 \\ -6\beta_1^3 & 6\beta_1\left(\beta_1 - 1\right)\left(\beta_1 + 1\right) & 6\beta_1 & 0 \\ 2\beta_1^3 & 4\beta_1\left(\beta_1 + 1\right) + \beta_2 & 2 & 0 \end{bmatrix} \tag{6.29}$$

where $\Delta = \left[\left(2\beta_1 + 4\right)\beta_1 + 4\right]\beta_1 + \beta_2 + 2$.

6. **Overhauser** spline basis, which depends on two parameters α and β

$$\mathbf{M}^{OV} = \begin{bmatrix} -\frac{(1-\alpha)^2}{\alpha} & \beta + \frac{1-\alpha}{\alpha} & -\frac{1-\alpha}{1-\beta} & \frac{\beta^2}{1-\beta} \\ \frac{2(1-\alpha)^2}{\alpha} & -\frac{2(1-\alpha)+\alpha\beta}{\alpha} & \frac{2(1-\alpha)-\beta(1-2\alpha)}{1-\beta} & -\frac{\beta^2}{1-\beta} \\ -\frac{(1-\alpha)^2}{\alpha} & \frac{1-2\alpha}{\alpha} & \alpha & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{6.30}$$

6.3 General properties of Spline functions

Spline functions satisfies a certain number of general properties which make them an interesting tool in many domains. It is helpful to introduce these properties in this point.

For modeling or interpolating a function, we can use polynomial models. However, if the order of the polynomial is too small, it will not result in a good approximation for the rapid varying parts of the function. Conversely, if the order is chosen too large, the estimated function may be too varying in the other points. In other words, as shown in [58]

Proposition 1 *In a polynomial approximation, if the function to be approximated is badly behaved anywhere in the interval of approximation, then the approximation is poor everywhere.*

However, by using low order splines (2nd degree or 3rd degree piecewise polynomial functions) we can well approximate rapid varying parts of a function (provided that there is enough knot points in that region), without affecting the other parts of the function.

Another nice property of the splines, is the minimum curvature property of the cubic splines [4, 83], expressed by the following:

Theorem 16 (Holladay) *Let the mesh $a = x_0 < x_1 < \ldots < x_N = b$ and a set of real numbers $\{y_i\}^N{}_{i=0}$ are given. Among all the functions $f(x)$ with a continuous second order derivative on $[a, b]$ and $f(x_i) = y_i$ $(i = 0, \ldots, N)$, the cubic spline $sp(x)$ with the knot sequence $\{x_i\}$ and the end conditions $sp''(a) = sp''(b) = 0$ is the function which minimizes the integral:*

$$\int_a^b \left|f''(x)\right|^2 dx$$

Proposition 2 *Any k-th order spline with the knot sequence $\xi = \{\xi_i\}^{l+1}{}_1$ can be represented as the sum:*

$$sp(x) = a_1 B_1(x) + a_2 B_2(x) + \ldots + a_N B_N(x)$$

where $B_i(x)$ are some predetermined and fixed splines which are determined only by knowing the knot sequence. In other words, $\{B_i(x)\}$ are the basis functions of the linear space of the splines with a known knot sequence.

Because of this property, the splines $B_i(x)$ are usually called *B-splines*.

This property also shows that if we model a function on an interval (with some predetermined knot sequence), the model is a linear model with respect to the parameters of the model (i.e. the coefficients a_i).

Proposition 3 *For a CR-spline the curve tangency lines, at the points Q_{i+1} and Q_{i+2}, are parallel to the straight lines passing through the points Q_i and Q_{i+2} and through the points Q_{i+1} and Q_{i+3}, respectively (see Fig. 6.6)*

Proof. Doing the derivatives of (6.20) with respect of ν, we obtain

$$\begin{aligned}\left.\frac{\partial y}{\partial \nu}\right|_{\nu=0} &= \tfrac{1}{2}\left(-Q_i + Q_{i+2}\right)\\ \left.\frac{\partial y}{\partial \nu}\right|_{\nu=0} &= \tfrac{1}{2}\left(-Q_{i+1} + Q_{i+3}\right)\end{aligned}$$

This justifies the property, as can be seen from Figure 6.6. ■

Proposition 4 *The spline function and its derivative are continuous in their control points Q_i.*

Proof. For the i-th span $F_i(\nu)$ it is easily to show that

$$\begin{aligned}\lim_{\nu\to 0} F_i(\nu) &= \lim_{\nu\to 1} F_{i-1}(\nu)\\ \lim_{\nu\to 1} F_i(\nu) &= \lim_{\nu\to 0} F_{i+1}(\nu)\\ \lim_{\nu\to 0} \tfrac{\partial F_i(\nu)}{\partial \nu} &= \lim_{\nu\to 1} \tfrac{\partial F_{i-1}(\nu)}{\partial \nu}\\ \lim_{\nu\to 1} \tfrac{\partial F_i(\nu)}{\partial \nu} &= \lim_{\nu\to 0} \tfrac{\partial F_{i+1}(\nu)}{\partial \nu}\end{aligned}$$

■

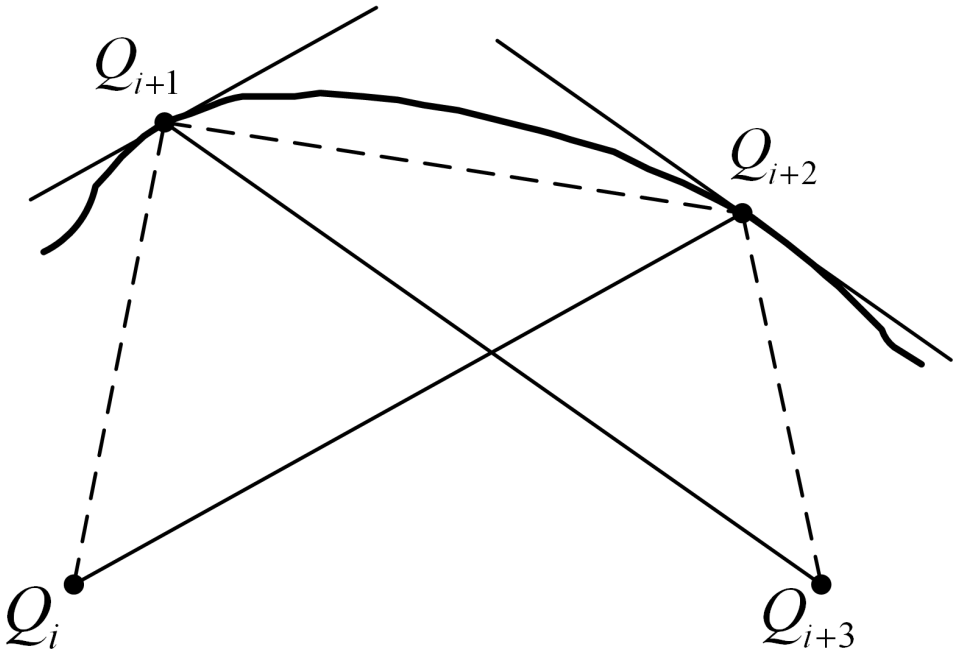

Fig. 6.6: Spline property in proposition 3

6.4 Spline implementation of the Complex Activation Function

The new idea is to use spline-based functions for $h^R(\bullet)$ and $h^I(\bullet)$ in eq. (5.3). Thus, remembering the spline matrix formulation in eq. (6.20) we can use two mono-dimensional spline functions. First we calculate the indexes span i_R and i_I, and the local parameters ν_R and ν_I, as stated by equations (6.23), then the AF in eq. (5.3) can be written as follows using eq. (6.20):

$$y_k = h_k^R(u_{Rk}) + jh_k^I(u_{Ik}) = \mathbf{T}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}_{i_R}^R + j\mathbf{T}(u_{Ik}) \cdot \mathbf{M} \cdot \mathbf{Q}_{i_I}^I \quad (6.31)$$

where $y_k = y_{Rk} + jy_{Ik}$ is the k-th complex output corresponding to the k-th input $u_k = u_{Rk} + ju_{Ik}$ of the k-th activation function, $\mathbf{Q}_{i_R}^R$ and $\mathbf{Q}_{i_I}^I$ collect the control points of the real and imaginary curve, respectively. The data path is reported in Figure 6.7.

This AF is known as *splitting activation function (SAF)* and the corresponding neural network is called *Complex valued Adaptive Spline Neural Network (CASNN)*.

Unfortunately the real and imaginary parts of a complex signal are usually correlated, not split in separate channels. In this way we need a better model of the complex AF. In this way, Vitagliano et al. in [214] proposed a complex neural network based on bi-dimensional spline AF.

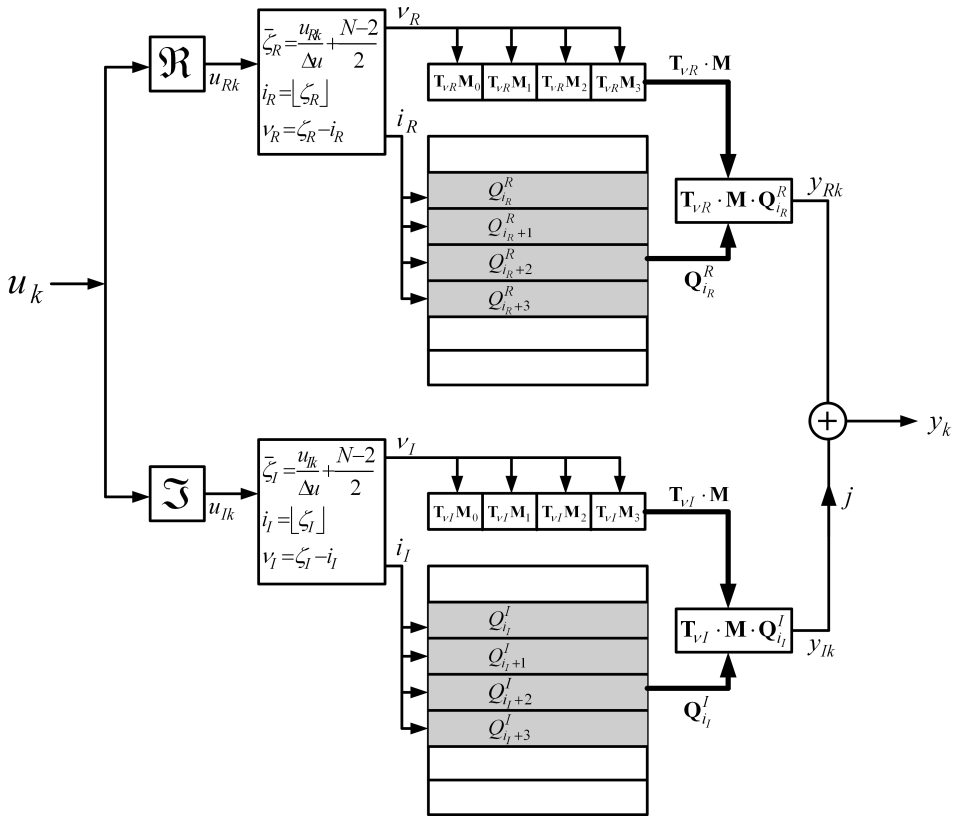

Fig. 6.7: Mono-dimensional spline data path of a complex activation function (AF)

If we consider the expression (5.4) of a complex function in relation to the real and imaginary part, we can render each of the two bi-dimensional real functions $h^R(u_R, u_I)$ and $h^I(u_R, u_I)$ with bi-dimensional splines: one plays the role of the real part and one the imaginary part of the complex activation function. This AF is known as *generalized splitting activation function (GSAF)* [214, 164].

Using the compact matrix formulation in eq. (6.24) we have for the k-th AF

$$\begin{aligned} y_{Rk} &= h^R_k(u_{Rk}, u_{Ik}) = \mathbf{T}_{\nu I}(u_{Ik}) \cdot \mathbf{M} \cdot (\mathbf{T}_{\nu R}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}^{(i_R,i_I)}_{[2]R})^T, \\ y_{Ik} &= h^I_k(u_{Rk}, u_{Ik}) = \mathbf{T}_{\nu I}(u_{Ik}) \cdot \mathbf{M} \cdot (\mathbf{T}_{\nu R}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}^{(i_R,i_I)}_{[2]I})^T, \\ y_k &= y_{Rk} + j y_{Ik}. \end{aligned} \tag{6.32}$$

Both the real and imaginary part of the k-th input signal u_k are evaluated by two flexible and bi-dimensional functions. The output of each function is real-valued; we impose these two outputs to be the real and the imaginary part of the output of the complex activation function respectively. The data path is reported in Figure 6.8.

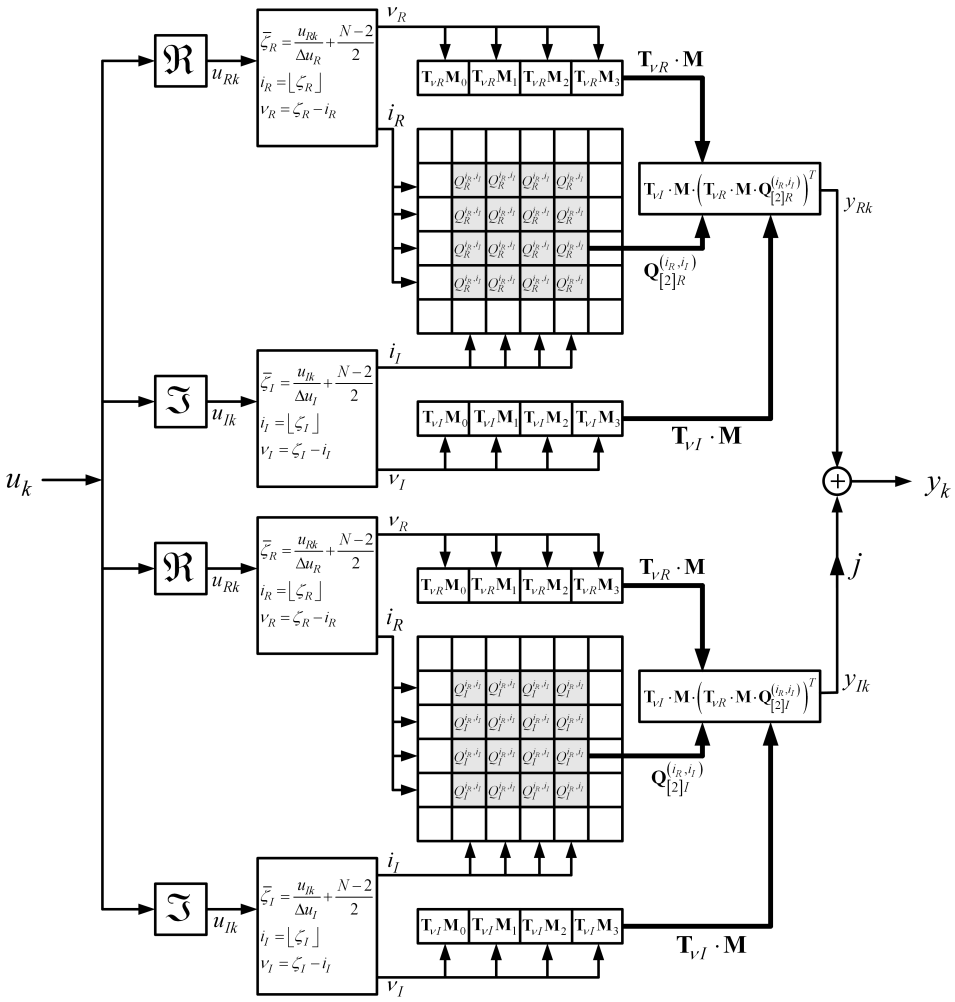

Fig. 6.8: Bi-dimensional spline data path of a complex activation function (AF)

7 The Flexible Solution

—We often think that when we have completed our study of one, we know all about two, because "two" is "one and one". We forget that we have still to make a study of "and".
Sir A. Eddington

EFFICIENT design of the de-mixing strategy requires the choice of a proper de-mixing model, a cost function able to measure the independence of the outputs and an effective optimization method. In this section a feed-forward neural network will be proposed and investigated as effective de-mixing model. Network parameters will be iteratively adapted (i.e. learned) on the basis of a measure of the output independence.

As said in Chapter 1 several approaches to blind separation of sources exist, but in this chapter we focus the attention on a set of algorithms which are based on the INFOMAX principle introduced by Bell and Sejnowski in [15]. This learning algorithm maximizes information transferred by the nonlinear network shown in Figure 7.1, assuming no knowledge on input vector distribution.

7.1 The INFOMAX algorithm

INFOMAX addresses the problem of maximizing the mutual information $I(\mathbf{y}, \mathbf{x})$ [55], between the input vector $\mathbf{x}$ and an invertible nonlinear transform of it, $\mathbf{y}$ obtained as

$$\mathbf{y} = \mathbf{h}(\mathbf{u}) = \mathbf{h}(\mathbf{W}\mathbf{x}) \tag{7.1}$$

where $\mathbf{W}$ is an $N \times N$ matrix and $\mathbf{h}(\mathbf{u}) = [h_1(u_1), \ldots, h_N(u_N)]^T$ is the nonlinear function vector (see the sixth chapter in [77]). Because the mapping

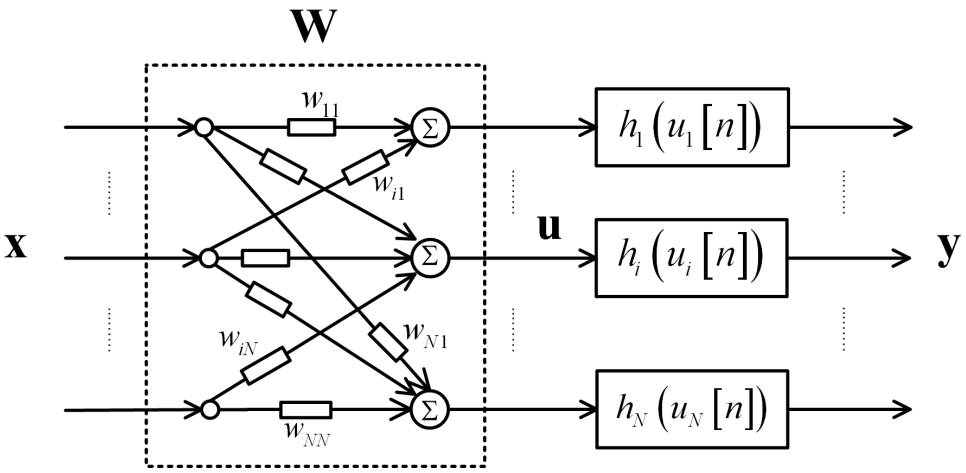

Fig. 7.1: The nonlinear network used to introduce the INFOMAX principle

in eq. (7.1) is deterministic, maximizing $I(\mathbf{y}, \mathbf{x})$ is the same that maximizing the joint entropy $H(\mathbf{y})$. In fact the following relation holds:

$$I(\mathbf{y}, \mathbf{x}) = H(\mathbf{y}) - H(\mathbf{y}|\mathbf{x}) \tag{7.2}$$

where $H(\mathbf{y}|\mathbf{x})$ is whatever entropy the output has which did not come from the input. In the case that we have no noise (or rather, we do not know what is noise and what is signal in the input), the mapping between $\mathbf{x}$ and $\mathbf{y}$ is deterministic and $H(\mathbf{y}|\mathbf{x})$ has its lowest possible value (it diverges to $-\infty$). This divergence is one of the consequences of the generalization of information theory to continuous variables. What we call $H(\mathbf{y})$ is really the "*differential*" entropy of $\mathbf{y}$ with respect to some reference, such as the noise level or the accuracy of our discretization of the variables in $\mathbf{x}$ and $\mathbf{y}$. The above equation can be differentiated as follows, with respect to a parameter, w involved in the mapping from $\mathbf{x}$ to $\mathbf{y}$:

$$\frac{\partial}{\partial w} I(\mathbf{y}, \mathbf{x}) = \frac{\partial}{\partial w} H(\mathbf{y}) \tag{7.3}$$

because $H(\mathbf{y}|\mathbf{x})$ does not depend on w.
In this way INFOMAX is equivalent to the entropy maximization.

The aim of INFOMAX algorithm is to adapt the entries of the matrix $\mathbf{W}$ maximizing the joint entropy $H(\mathbf{y})$ in (7.2). In order to derive the learning algorithm let we pose $p_{\mathbf{x}}(\mathbf{x})$ and $p_{\mathbf{y}}(\mathbf{y})$ the probability density functions (pdf) of the network input and output respectively which have to satisfy the relation [144]:

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{|\det \mathbf{J}|} \tag{7.4}$$

where $|\bullet|$ denotes the absolute value and $\mathbf{J}$ the Jacobian matrix of the transformation: $\mathbf{J} = [\partial y_i / \partial x_j]_{ij}$.

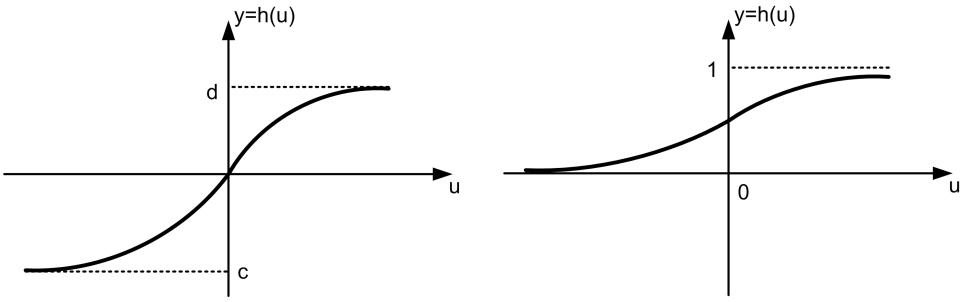

Fig. 7.2: Nonlinearities h_i used in the recovering network

The nonlinear transformations $h_i(u_i)$ are necessary for bounding the entropy in a finite range [218]. Indeed, when $h_i(u_i)$ is bounded $c \leq h_i(u_i) \leq d$ (see Figure 7.2), for any random variable u_i the entropy of $y_i = h_i(u_i)$ has an upper bound:

$$H(y_i) \leq \ln(d - c)$$

Therefore, the joint entropy of the transformed output vector is upper bounded:

$$H(\mathbf{y}) \leq \sum_{i=1}^{N} H(y_i) \leq N \ln(d - c) \tag{7.5}$$

In fact, the above inequality holds for any bounded transforms, so the global maximum of the entropy $H(\mathbf{y})$ exists. $H(\mathbf{y})$ may also have many local maxima determined by the functions h_i used to transform $\mathbf{u}$.

Since the joint entropy of network output is defined as $H(\mathbf{y}) = -E\{\ln p_{\mathbf{y}}(\mathbf{y})\}$ [55], where $E\{\bullet\}$ is the expected value operator, substituting into it the (7.4) we obtain:

$$H(\mathbf{y}) = E\{\ln |\det \mathbf{J}|\} + H(\mathbf{x}). \tag{7.6}$$

Now we can note that $\frac{\partial y_i}{\partial x_j} = \frac{\partial y_i}{\partial u_i}\frac{\partial u_i}{\partial x_j} = h_i'(u_i) \cdot w_{ij}$, so we obtain

$$\ln |\det \mathbf{J}| = \ln \det \mathbf{W} + \sum_{i=1}^{N} \ln \left|h_i'\right|. \tag{7.7}$$

Hence, the expression of the joint entropy $H(\mathbf{y})$ (ignoring the expected value operator $E\{\bullet\}$, replacing by instantaneous values) is:

$$H(\mathbf{y}) = H(\mathbf{x}) + \ln \det \mathbf{W} + \sum_{i=1}^{N} \ln \left|h_i'\right|. \tag{7.8}$$

The maximization (or minimization) of a generic cost function $\mathcal{L}\{\mathbf{\Phi}\}$ with respect a parameter $\mathbf{\Phi}$ can be obtained by the application of the stochastic gradient method at $(l+1)$-th iteration

$$\Phi(l+1) = \Phi(l) + \eta_\Phi \frac{\partial L\{\Phi(l)\}}{\partial \Phi} = \Phi(l) + \eta_\Phi \mathbf{\Delta\Phi}(l) \tag{7.9}$$

where $\eta_{\mathbf{\Phi}}$ is the learning rate.

Remembering that $H(\mathbf{x})$ is not affected by the parameters that we are learning, it is possible to write the learning rule for the matrix $\mathbf{w}$ using the stochastic gradient method in (7.9) as follows:

$$\Delta\mathbf{W} = \frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} = \mathbf{W}^{-T} + \mathbf{\Psi}\mathbf{x}^T \tag{7.10}$$

where $\mathbf{W}^{-T} = \left(\mathbf{W}^{-1}\right)^T$, $\mathbf{\Psi} = [\Psi_1, \ldots, \Psi_N]^T$ and $\Psi_k = {h_k''(u_k)}/{h_k'(u_k)}$.

But what is the relationship between INFOMAX and ICA?
For this scope we can introduce the mutual information of the linear outputs $\mathbf{u}$ as the Kullback-Leibler distance [55] of the output distribution $I(\mathbf{u}) = E\left\{\ln\left(p_{\mathbf{u}}(\mathbf{u}) \Big/ \prod_{i=1}^N p_{u_i}(u_i)\right)\right\}$, where $p_{\mathbf{u}}(\mathbf{u})$ is the joint pdf of the output vector $\mathbf{u}$ and $p_{u_i}(u_i)$ are the marginal pdfs. Using this relation and the (7.7) in the definition of entropy, after some easy passes:

$$\begin{aligned}
&H(\mathbf{y}) = -\int p_{\mathbf{y}}(y) \ln p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y} = -E\{\ln p_{\mathbf{y}}(\mathbf{y})\} = \\
&= -E\left\{\ln \frac{p_{\mathbf{u}}(\mathbf{u})}{\prod_{i=1}^N |h_i'|}\right\} = -E\{\ln p_{\mathbf{u}}(\mathbf{u})\} + E\left\{\ln \prod_{i=1}^N |h_i'|\right\} = \\
&= -E\{\ln p_{\mathbf{u}}(\mathbf{u})\} + E\left\{\ln \prod_{i=1}^N p_{u_i}(u_i)\right\} - E\left\{\ln \prod_{i=1}^N p_{u_i}(u_i)\right\} + E\left\{\ln \prod_{i=1}^N |h_i'|\right\} = \\
&= -E\left\{\ln \frac{p_{\mathbf{u}}(\mathbf{u})}{\prod_{i=1}^N p_{u_i}(u_i)}\right\} + E\left\{\sum_{i=1}^N \ln \frac{|h_i'|}{p_{u_i}(u_i)}\right\} = \\
&= -I(\mathbf{u}) + E\left\{\sum_{i=1}^N \ln \frac{|h_i'|}{p_{u_i}(u_i)}\right\}
\end{aligned}$$

we obtain:

$$H(\mathbf{y}) = -I(\mathbf{u}) + E\left\{\sum_{i=1}^N \ln \frac{|h_i'|}{p_{u_i}(u_i)}\right\}. \tag{7.11}$$

Thus if $|h_i'| = p_{u_i}(u_i)$ $(\forall i)$ then maximizing the joint entropy $H(\mathbf{y})$ is *equivalent* to minimizing the mutual information, that is the Kullback-Leibler divergence (which is a measure of the independence of the u_i signals) and so the ICA problem is solved. In this way $h_i(u_i)$ should be the cumulative density function (cdf) of the i-th estimated source. The use of an adaptive AF can successfully fulfills the matching of $h_i(u_i)$ to the cdf of the i-th source [166].

Moreover remembering the eq. (7.5), the use of a cdf-like function for the h_i functions allows the joint entropy $H(\mathbf{y})$ to have in $H(\mathbf{y}) = \mathbf{0}$ its global maximum (because $c = 0$ and $d = 1$), see Figure 7.2.

From (7.11) the INFOMAX algorithm can be performed by two equivalent approach: maximizing the joint entropy of the network output (*ME approach*) or minimizing the mutual information (*MMI approach*) [218, 191].

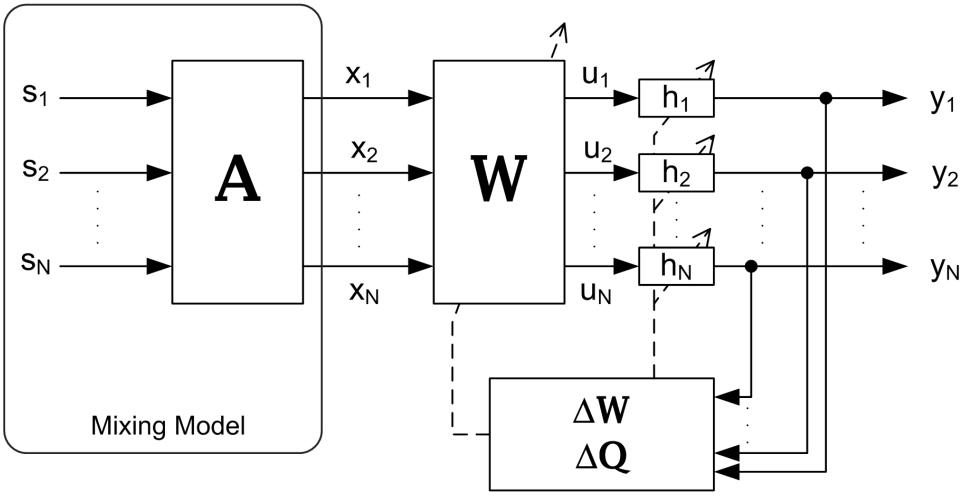

Fig. 7.3: The overall model in the linear case

Our approach performs separation by maximization of the joint entropy of the network outputs $\mathbf{y}$, extending the conventional real-domain *INFOMAX algorithm* to the complex domain [26]. The choice of the ME approach is supported by the fact that the joint entropy is an intuitively meaningful contrast function (an objective function for source separation, which measures the statistical independence), it usually allows simple learning rules and it is closely related to several other approaches [111].

7.2 The demixing algorithm in the linear environment

The overall system is shown in Figure 7.3, while the architecture used to realize the model $\mathbf{W}$ in (4.2) is particularized in Figure 7.4.

Let we consider first the case of using the splitting activation function in (5.3) realized through the mono-dimensional spline activation function in (6.31) [166], assuming

$$\begin{aligned} \mathbf{y} &= \mathbf{h}\left(\mathbf{u}\right) = \mathbf{h}_R(u_R) + j\mathbf{h}_I(\mathbf{u}_I) \\ y_k &= y_{Rk} + jy_{Ik} = h_{Rk}(u_{Rk}) + jh_{Ik}(u_{Ik}) \end{aligned} \tag{7.12}$$

where $\mathbf{h}$ is the activation function vector and y_k is the k-th element in $\mathbf{y}$, the expression of the complex output vector $\mathbf{y}$ can be rewritten by using only real terms:

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_R\left[n\right] \\ \mathbf{y}_I\left[n\right] \end{bmatrix} = \begin{bmatrix} \mathbf{h}_R\left(\mathbf{u}_R\left[n\right]\right) \\ \mathbf{h}_I\left(\mathbf{u}_I\left[n\right]\right) \end{bmatrix}. \tag{7.13}$$

In this way $\tilde{\mathbf{y}}$ is a real vector of $2N$ elements. Considering a de-mixing model with parameters $\Phi = \{w_{ij}, \Phi_h \,|\forall i,j\,\}$, where w_{ij} are the entries in matrix $\mathbf{W}$ and $\mathbf{\Phi}_h = \left\{Q_R^h, Q_I^h\right\}$ are the spline control points for the real and imaginary

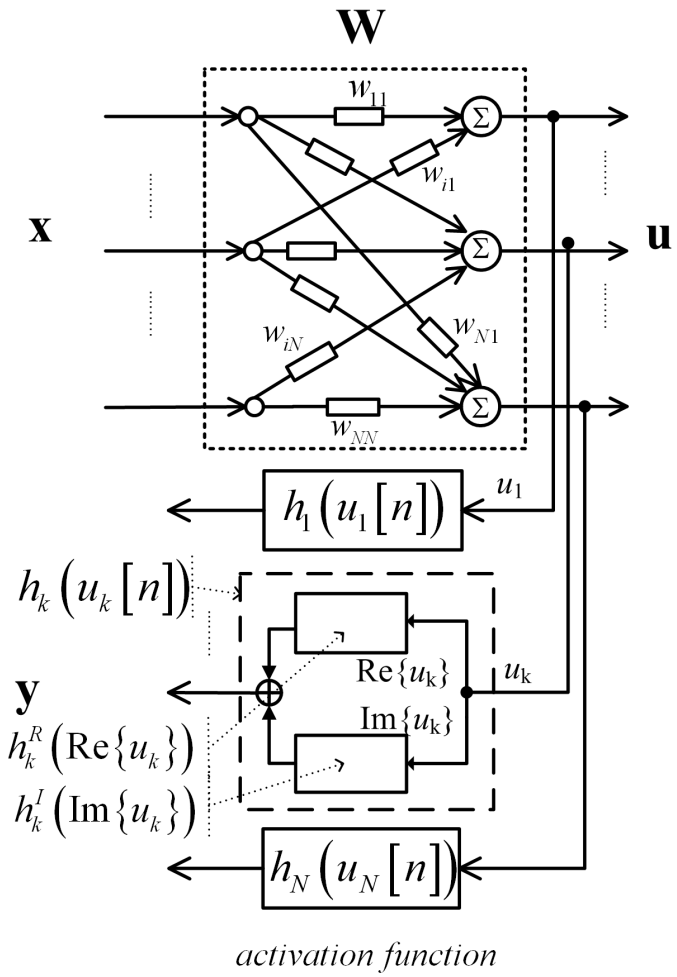

Fig. 7.4: De-mixing model in linear environment. The activation functions are realized with splitting functions

part of the AF, the cost function to be maximized is the joint entropy of the signals after the activation functions, similarly to the eq. (7.6):

$$\mathcal{L}\left\{\mathrm{y}\left[n\right],\Phi\right\} = H\left(\tilde{\mathbf{y}}\right) = -E\left\{\ln\left(p_{\tilde{\mathbf{y}}}\left(\tilde{\mathbf{y}}\right)\right)\right\} = H\left(\tilde{\mathbf{x}}\right) + E\left\{\ln\left(\tilde{\mathbf{J}}\right)\right\}. \quad (7.14)$$

In eq. (7.14) the output pdf $p_{\tilde{\mathbf{y}}}\left(\tilde{\mathbf{y}}\right)$ can be expressed using eq. (7.4) as a function of the model's parameters and of $H(\tilde{\mathbf{x}})$ which does not depend on the model's parameter. In this case the Jacobian of the transformation between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ can be expressed as follows:

$$\det\left(\tilde{\mathbf{J}}\right) = \prod_{k=1}^{N} \dot{\tilde{y}}_k \det\left(\tilde{\mathbf{W}}\right). \quad (7.15)$$

In eq. (7.15) $\dot{\tilde{y}}_k$ is the derivative of the k-th elements of $\tilde{\mathbf{y}}$.

Having explored the mixing model, the associated cost function and the recovering network, the next step is to derive the learning rules. Substituting eq. (7.15) in eq. (7.14) we obtain the cost function whose derivation with respect the elements of $\mathbf{\Phi}$ leads to

$$\frac{\partial}{\partial \Phi}\mathcal{L}\left\{\mathrm{y}, \Phi\right\} = \frac{\partial}{\partial \Phi}\left[\ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) + \sum_{k=1}^{N}\ln \dot{y}_{Rk} + \sum_{k=1}^{N}\ln \dot{y}_{Ik}\right] \tag{7.16}$$

where $\dot{\tilde{y}}_{Rk}$ and $\dot{\tilde{y}}_{Ik}$ denote the derivative of the k-th real and imaginary parts of the network output $\mathbf{y}$. In eq. (7.16) expected values have been replaced by instantaneous values.

Maximization of eq. (7.16) by the stochastic gradient method in eq. (7.9) yields three learning rules (see [200] for major details). The learning rule for the network's weights is:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{x}^H \tag{7.17}$$

where H is the Hermitian operator, $\mathbf{W}^{-H} = \left(\mathbf{W}^{-1}\right)^H$, $\mathbf{\Psi} = \mathbf{\Psi}_R + j\mathbf{\Psi}_I$, $\psi_R = [\Psi_{R1}, \ldots, \Psi_{RN}]^T$, $\psi_I = [\Psi_{I1}, \ldots, \Psi_{IN}]^T$, $\Psi_{Rk} = \ddot{y}_{Rk}/\dot{y}_{Rk}$, $\Psi_{Ik} = \ddot{y}_{Ik}/\dot{y}_{Ik}$, $\dot{y}_{Rk}$ is the derivative of the k-th real part element of $\mathbf{y}$ while $\ddot{y}_{Rk}$ is the second order derivative and similarly for the imaginary counterpart $\dot{y}_{Ik}$ and $\ddot{y}_{Ik}$. Using the matrix notation in eq. (6.20) the terms Ψ_{Rk} and Ψ_{Ik} can be expressed as follows:

$$\begin{aligned}\Psi_{Rk} &= \frac{1}{\Delta u_R}\frac{\ddot{\mathbf{T}}_{\nu R}\mathbf{M}\mathbf{Q}_R^h}{\dot{\mathbf{T}}_{\nu R}\mathbf{M}\mathbf{Q}_R^h},\\ \Psi_{Ik} &= \frac{1}{\Delta u_I}\frac{\ddot{\mathbf{T}}_{\nu I}\mathbf{M}\mathbf{Q}_I^h}{\dot{\mathbf{T}}_{\nu I}\mathbf{M}\mathbf{Q}_I^h}\end{aligned} \tag{7.18}$$

where $\dot{\mathbf{T}}_{\nu R} = \left[\begin{array}{cccc} 3\nu_R^2 & 2\nu_R & 1 & 0\end{array}\right]$, $\ddot{\mathbf{T}}_{\nu R} = \left[\begin{array}{cccc} 6\nu_R & 2 & 0 & 0\end{array}\right]$ and similar for the imaginary counterpart $\dot{\mathbf{T}}_{\nu I}$ and $\ddot{\mathbf{T}}_{\nu I}$.

The learning rules for the spline activation functions are:

$$\begin{aligned}\Delta\mathbf{Q}_{\mathrm{R},k,i+m}^h &= \frac{\dot{\mathbf{T}}_{\nu R}(u_{\mathrm{R}k})(\mathbf{M})_m}{\dot{\mathbf{T}}_{\nu R}(u_{\mathrm{R}k})\mathbf{M}\mathbf{Q}_{\mathrm{R},k,i+m}^h},\\ \Delta\mathbf{Q}_{\mathrm{I},k,i+m}^h &= \frac{\dot{\mathbf{T}}_{\nu I}(u_{\mathrm{I}k})(\mathbf{M})_m}{\dot{\mathbf{T}}_{\nu I}(u_{\mathrm{I}k})\mathbf{M}\mathbf{Q}_{\mathrm{I},k,i+m}^h}\end{aligned} \tag{7.19}$$

where $(\mathbf{M})_m$ is the m-th column of the $\mathbf{M}$ matrix.

7.2.1 The use of the generalized splitting function

We can generalize this algorithm using the generalized splitting function in eq. (5.4) realizing the complex AFs with the *bi-dimensional spline function* in eq. (6.32) [164, 165, 166] shown in Figure 7.5. In this case the algorithm is formally very similar to the previous case: the learning rule for the matrix

weights w_{ij} is formally identical to eq. (7.17), but the generic k-th terms Ψ_{Rk} and Ψ_{Ik} of the vector $\mathbf{\Psi}$ contain the partial and cross derivatives of network outputs y_{Rk} and y_{Ik} (outputs of bi-dimensional functions) with respect the two variables u_{Rk} and u_{Ik}:

$$\begin{cases} \psi_{iR} = 2\dfrac{\frac{\partial y_{iR}}{\partial u_{iR}}\frac{\partial^2 y_{iR}}{\partial u_{iR}^2}+\frac{\partial y_{iR}}{\partial u_{iI}}\frac{\partial}{\partial u_{iR}}\frac{\partial y_{iR}}{\partial u_{iI}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2+\left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} \\ \psi_{iI} = 2\dfrac{\frac{\partial y_{iR}}{\partial u_{iI}}\frac{\partial^2 y_{iR}}{\partial u_{iI}^2}+\frac{\partial y_{iR}}{\partial u_{iR}}\frac{\partial}{\partial u_{iI}}\frac{\partial y_{iR}}{\partial u_{iR}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2+\left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} \end{cases} \tag{7.20}$$

Using the matrix notation in eq. (6.24) the terms Ψ_{Rk} can be expressed as

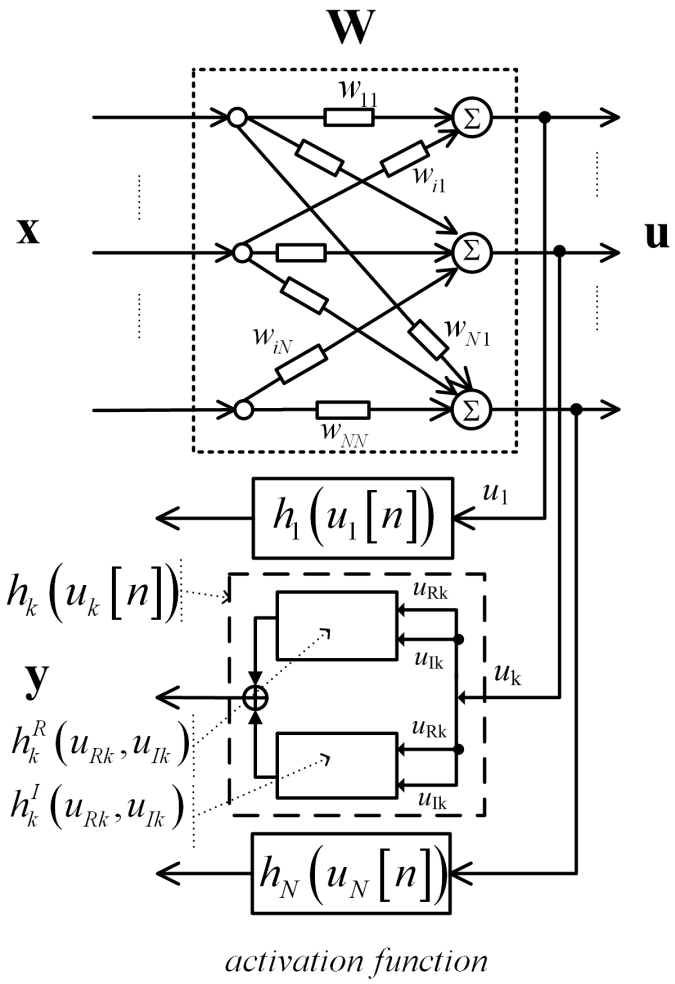

Fig. 7.5: De-mixing model in linear environment. The activation functions are realized with generalized splitting functions

follows:

$$\psi_{iR} = \frac{2}{\Delta}\left(\frac{\left(\mathbf{T}_{Ii}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)\left(\mathbf{T}_{Ii}\cdot\mathbf{M}\cdot\left(\ddot{\mathbf{T}}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)}{\left(\mathbf{T}_{Ii}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{Ii}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)^2} + \right.$$
$$\left. + \frac{\left(\dot{\mathbf{T}}_{Ii}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)\left(\dot{\mathbf{T}}_{Ii}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)}{\left(\mathbf{T}_{Ii}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{Ii}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{Ri}\cdot\mathbf{M}\cdot\mathbf{Q}_{Ri}\right)^T\right)^2}\right) \tag{7.21}$$

and similarly for Ψ_{Ik}.

The learning rule for the real activation function becomes

$$\Delta\mathbf{Q}^h_{\mathrm{R},k,i_R+m_R,i_I+m_I} = \frac{\partial \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2+\left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} =$$
$$= \begin{cases} 0 & i \neq j \\ 2\dfrac{\frac{\partial y_{jR}}{\partial u_{jR}}\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial u_{jR}}+\frac{\partial y_{jR}}{\partial u_{jJ}}\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial u_{jI}}}{\left(\frac{\partial y_{jR}}{\partial u_{jR}}\right)^2+\left(\frac{\partial y_{jR}}{\partial u_{jI}}\right)^2} & i = j \end{cases} \tag{7.22}$$

Using the matrix notation in eq. (6.24) we obtain

$$\Delta\mathbf{Q}^h_{\mathrm{R},k,i_R+m_R,i_I+m_I} = 2\left(\frac{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}_{m_J}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}_{m_R}\right)^T\right)}{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)^2} + \right.$$
$$\left. + \frac{\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}_{m_J}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}_{m_R}\right)^T\right)}{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^T\right)^2}\right) \tag{7.23}$$

where $\mathbf{M}_k$ is a matrix in which all the elements are zero, except the k-th column, which is equal to the k-th column of the matrix $\mathbf{M}$. A similar equation can be resulted for the imaginary surface $\Delta\mathbf{Q}^h_{\mathrm{I},k,i_R+m_R,i_I+m_I}$. For a complete derivation of the learning rules see appendix C.1 and [165].

7.3 The choice of the de-mixing model in the nonlinear case

In designing the de-mixing model in the nonlinear environment, it is important to find the theoretical conditions in terms of sources, mixing environment and recovering architecture capable of guaranteeing the existence of the solution [183].

The model of nonlinear complex compensating functions $\mathbf{G}\left(\mathbf{x}\left[n\right]\right)$ and the activation functions (AF) considered in this chapter are realized by splitting function according to (5.3) (technical details on the implementation of splitting functions will be given in the following sections).

Let $\mathcal{U}$ be the set of all complex vectors $\mathbf{u}$ with joint pdf $p_{\mathbf{u}}(\mathbf{u})$ (see Figure 2.4) having independent components u_i with marginal pdf $p_{u_i}(u_i)$ [166]

$$\mathcal{U} = \left\{ \mathbf{u} \left| p_{\mathbf{u}}(\mathbf{u}) = \prod_i p_{u_i}(u_i);\ \mathbf{u} = \mathcal{G}\{\mathbf{x}\} = \mathcal{G} \circ \mathcal{F}\{\mathbf{s}\} = \mathcal{H}\{\mathbf{s}\} \right. \right\} \quad (7.24)$$

where $\mathcal{H}(\bullet)$ is an unspecified application with a non-diagonal Jacobian matrix in general. As a matter of fact, it is possible to find an infinite number of models $\mathcal{G}(\bullet)$ such that $\mathbf{u} = \mathcal{G}\{\mathbf{x}\} \in \mathcal{U}$, but not all of them have a diagonal Jacobian matrix. So most of the solutions in $\mathcal{U}$ are not of interest, meaning that output independence by itself is a weak approach to the BSS problem in a general nonlinear environment.

Considering the splitting realization of the complex nonlinear distorting function $\mathbf{F}(\mathbf{v}[n]) = \mathbf{F}_R(\mathbf{v}_R[n]) + j\mathbf{F}_I(\mathbf{v}_I[n])$, $\mathbf{F}_R(\mathbf{v}_R[n]) = [f_{R1}(v_{R1}[n]), \ldots, f_{RN}(v_{RN}[n])]^T$ and $\mathbf{F}_I(\mathbf{v}_I[n]) = [f_{I1}(v_{I1}[n]), \ldots, f_{IN}(v_{IN}[n])]^T$, the complex domain mixing environment in (4.3), represented in [211, 212], can be rewritten in the following way:

$$\begin{aligned} \tilde{\mathbf{x}}[n] &= \begin{bmatrix} \mathbf{x}_R[n] \\ \mathbf{x}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{F}_R(\mathbf{v}_R[n]) \\ \mathbf{F}_I(\mathbf{v}_I[n]) \end{bmatrix}, \\ \tilde{\mathbf{v}}[n] &= \begin{bmatrix} \mathbf{v}_R[n] \\ \mathbf{v}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix} \begin{bmatrix} \mathbf{s}_R[n] \\ \mathbf{s}_I[n] \end{bmatrix} = \tilde{\mathbf{A}}\tilde{\mathbf{s}} \end{aligned} \quad (7.25)$$

where $\mathbf{A}_R$ and $\mathbf{A}_I$ are the real and imaginary parts of the complex mixing matrix $\mathbf{A} = \mathbf{A}_R + j\mathbf{A}_I$.

Equations (7.25) have the very attractive property of involving only real quantities, thus making it possible to convert complex mixing models into real models of increased size.

It is now possible to define the de-mixing models and to design the network performing the source separation. In particular, a priori knowledge about the mixing model is exploited to design the recovering network. So the *mirror model* in eq. (4.4) has been introduced to grant the existence and the uniqueness of the solution (up to the trivial indeterminacy of the ICA approach to BSS) as described in [208, 212].

The nonlinear complex compensating functions $\mathbf{G}(\bullet)$ have been realized as splitting functions according to eq. (5.3): $\mathbf{G}(\mathbf{x}_R[n] + j\mathbf{x}_I[n]) = \mathbf{G}_R(\mathbf{x}_R[n]) + j\mathbf{G}_I(\mathbf{x}_I[n])$.

Similarly to eq. (7.25) it is possible to express the complex de-mixing model in eq. (4.4) by using real expressions only:

$$\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_R[n] \\ \mathbf{u}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \mathbf{G}_R(\mathbf{x}_R[n]) \\ \mathbf{G}_I(\mathbf{x}_I[n]) \end{bmatrix} = \tilde{\mathbf{W}} \cdot \tilde{\mathbf{G}}[\tilde{\mathbf{x}}] \quad (7.26)$$

in which $\mathbf{G}_R(\mathbf{x}_R[n]) = [g_{R1}(x_{R1}[n]), \cdots, g_{RN}(x_{RN}[n])]^T$ and $\mathbf{G}_I(\mathbf{x}_I[n]) = [g_{I1}(x_{I1}[n]), \cdots, g_{IN}(x_{IN}[n])]^T$ are the real and imaginary parts of the nonlinear compensating functions while $\mathbf{W}_R$ and $\mathbf{W}_I$ are the real and imaginary

parts of the complex mixing matrix $\mathbf{W} = \mathbf{W}_R + j\mathbf{W}_I$. Equation (7.26) represents a real-valued PNL model and preserves all the properties of PNL BSS in the real domain. In particular it is possible to extend to the complex domain the results of [186] for the real PNL case (already applied in [208] for the real convolutive PNL mixture, also known as C-PNL), specifically the proof of existence and uniqueness of the solution.

For the problem herein considered, elements of set $\mathcal{U}$, under proper constraints, differ only for a trivial ambiguity if the mixing model is eq. (4.3) and the de-mixing model is eq. (4.4). This is shown in the following theorem [212, 166].

Theorem 17 *Given the nonlinear complex mixing model $\mathcal{F}\{\mathbf{A}, \mathbf{F}\}$ in eq.* (4.3) *and the recovery model $\mathcal{G}\{\mathbf{G}, \mathbf{W}\}$ in eq.* (4.4)*, let us assume that:*

a $\mathbf{A}$ *is a non-singular matrix of non zero entries (both for real and imaginary part) and not absolutely degenerate;*

b $f_{Ri}(\cdot), f_{Ii}(\cdot), \mathrm{g}_{Ri}(\cdot), \mathrm{g}_{Ii}(\cdot)$ *($i = 1, \ldots, N$) are diagonal, differentiable, invertible and zero preserving monotonic functions;*

c $\mathbf{s}[n] = \mathbf{s}_R[n] + j\mathbf{s}_I[n]$ *is a complex random vector in which $\mathbf{s}_R[n], \mathbf{s}_I[n] \in \mathbb{R}$. The components of $\mathbf{s}[n]$ are statistically independent and have finite support;*

d the pdf of $s_i[n]$ ($i = 1, \ldots, N$) vanishes for at least one complex component, i.e. $i = l$.

Then the components of the output vector $\mathbf{u}[n] = \mathbf{u}_R[n] + j\mathbf{u}_I[n]$ are independent if and only if:

$$\mathbf{u}[n] = \mathbf{P}\mathbf{\Lambda}\mathbf{s}[n] = \mathbf{P} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix} \mathbf{s}[n] \tag{7.27}$$

In eq. (7.27) $\mathbf{P}$ *is a real permutation matrix and $\mathbf{\Lambda}$ is a complex diagonal matrix such that each element can be only purely real or imaginary.*

Proof. See Appendix C.2. ■

Theorem 17 ensures the existence and uniqueness of the solution at the expense of strong constraints on the real and the imaginary parts of the signals.

7.4 The demixing algorithm in the nonlinear environment

The overall system is shown in Figure 7.6 while the architecture used to realize the model $\mathcal{G}\{\mathbf{G}, \mathbf{W}\}$ in eq. (4.4) is particularized in Figure 7.7.

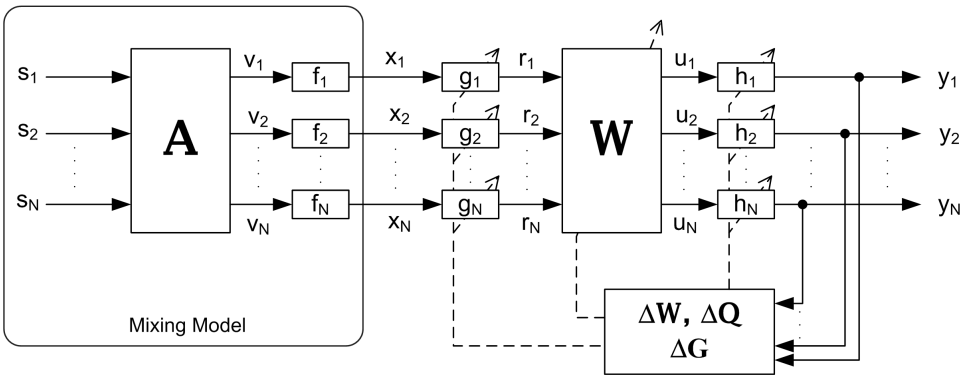

Fig. 7.6: The overall model in PNL case

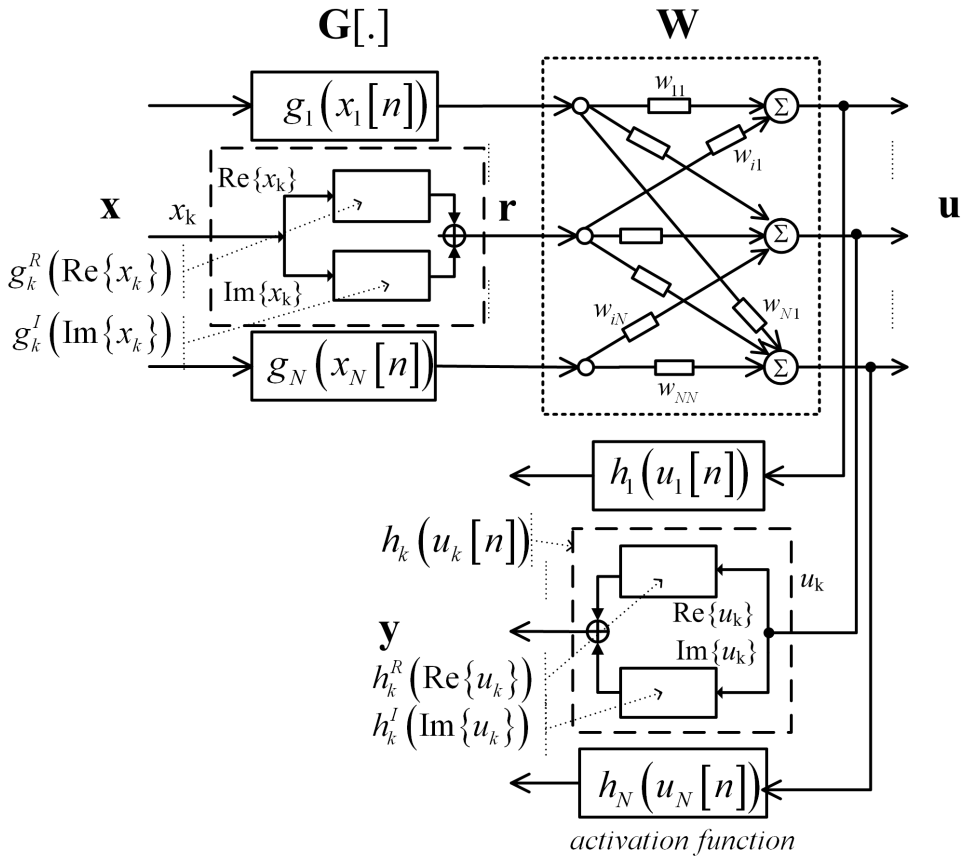

Fig. 7.7: De-mixing model in nonlinear environment

7.4.1 The ME algorithm

In the same way as the case of linear environment the de-mixing algorithm is based on an extension of the INFOMAX algorithm [15], performing

the ME approach.

The network output $\mathbf{y}$ is similar to eq. (7.12) [210, 209, 166]. The network parameters are $\Phi = \{w_{ij}, \Phi_h, \Phi_G \,|\forall i,j\}$, where w_{ij} are the entries in matrix $\mathbf{W}$, $\mathbf{\Phi}_h = \left\{Q_R^h, Q_I^h\right\}$ are the spline control points for the real and imaginary part of the AF and $\mathbf{\Phi}_G = \left\{Q_R^G, Q_I^G\right\}$ are the spline control points for the real and imaginary part of the nonlinear compensating functions $\mathbf{G}(\bullet)$. So using eq. (7.13) the cost function $\mathcal{L}\left\{\mathbf{y}\left[n\right], \mathbf{\Phi}\right\}$ to be maximized is the joint entropy in eq. (7.14).

In this case the Jacobian of the transformation between $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ can be expressed as follows:

$$\det\left(\tilde{\mathbf{J}}\right) = \prod_{k=1}^{N} \dot{\tilde{y}}_k \dot{\tilde{r}}_k \det\left(\tilde{\mathbf{W}}\right). \tag{7.28}$$

In eq. (7.28) $\dot{\tilde{y}}_k$ and $\dot{\tilde{r}}_k$ are the derivative of the k-th element of $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{r}}$ respectively.

Inserting eq. (7.28) in the derivation of the cost function eq. (7.14) with respect to the elements of parameters $\mathbf{\Phi}$ we obtain

$$\frac{\partial}{\partial\Phi}\mathcal{L}\left\{\mathrm{y}, \Phi\right\} = \frac{\partial}{\partial\Phi}\left[\ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) + \sum_{k=1}^{N}\ln\dot{y}_{Rk} + \sum_{k=1}^{N}\ln\dot{y}_{Ik} + \sum_{k=1}^{N}\ln\dot{r}_{Rk} + \sum_{k=1}^{N}\ln\dot{r}_{Ik}\right]. \tag{7.29}$$

In eq. (7.29) expected values have been replaced by instantaneous values.

Maximization of eq. (7.29) by the stochastic gradient method eq. (7.9) yields to five learning rules (see [212] for major details). The learning rule [210, 209, 166] for the network's weights is:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{r}^H \tag{7.30}$$

that is formally identical to eq. (7.17) and $\mathbf{\Psi}$ is defined analogously. The learning rules for the spline activation functions are:

$$\begin{aligned}\Delta\mathbf{Q}^h_{\mathrm{R},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{R})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{R})\mathbf{M}\mathbf{Q}^h_{\mathrm{R},k,i+m}},\\ \Delta\mathbf{Q}^h_{\mathrm{I},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{I})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{I})\mathbf{M}\mathbf{Q}^h_{\mathrm{I},k,i+m}}\end{aligned} \tag{7.31}$$

where $(\mathbf{M})_m$ is a vector composed by the m-th column of the matrix $\mathbf{M}$. The learning rules for the spline compensating functions are:

$$\begin{aligned}\Delta\mathbf{Q}^G_{\mathrm{R},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{R})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{R})\mathbf{M}\mathbf{Q}^G_{\mathrm{R},k,i+m}} + \mathrm{Re}\left\{\mathbf{\Psi}\left(\mathbf{W}^H\right)_k\right\}\tfrac{1}{2}\mathbf{T}_\nu\left(u_\mathrm{R}\right)\left(\mathbf{M}\right)_m,\\ \Delta\mathbf{Q}^G_{\mathrm{I},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{I})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{I})\mathbf{M}\mathbf{Q}^G_{\mathrm{I},k,i+m}} + \mathrm{Im}\left\{\mathbf{\Psi}\left(\mathbf{W}^H\right)_k\right\}\tfrac{1}{2}\mathbf{T}_\nu\left(u_\mathrm{I}\right)\left(\mathbf{M}\right)_m\end{aligned} \tag{7.32}$$

where $(\mathbf{W})_m$ is a vector composed by the k-th column of the matrix $\mathbf{W}$, $\mathrm{Re}\left\{\bullet\right\}$ and $\mathrm{Im}\left\{\bullet\right\}$ are the operators that return the real and imaginary parts of their inputs, respectively. For a complete derivation of the learning rules see Appendix C.3 and [212].

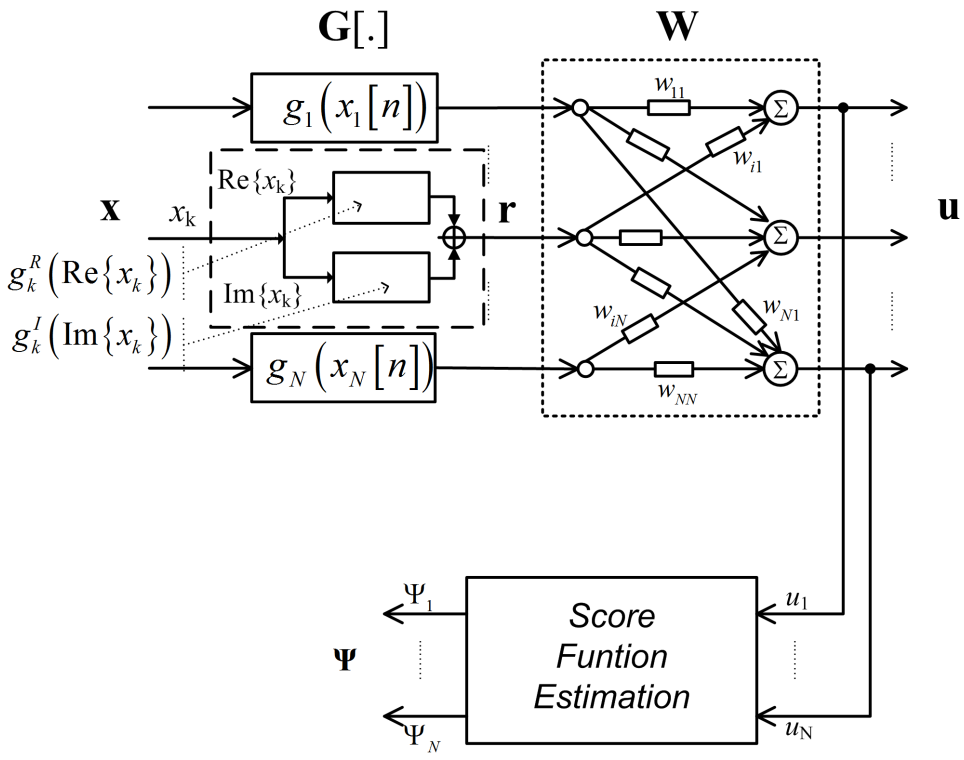

Fig. 7.8: The demixing model for the MMI algorithm

7.4.2 The MMI algorithm

An alternative algorithm can be obtained using the MMI approach deducing similar learning rules. This approach is presented in [211] and Appendix C.4.

The de-mixing model in nonlinear case and using the MMI approach is shown in Figure 7.8, where the differences with the ME approach are clear: the presence of the "*Score Function Estimation*" block and the absence of the Activation function.

The cost function is the Mutual Information that is the Kullback-Leibler divergence between the joint pdf $p_{\mathbf{u}}(\mathbf{u})$ and the product of its marginal pdfs $\prod_{i=1}^{N} p_{u_i}(u_i)$ (see eq. (7.11)). In this way we can measure the statistical independence directly. Therefore the cost function is

$$\begin{aligned}\mathcal{L}\{\tilde{\mathbf{u}}[n], \mathbf{\Phi}\} = I(\tilde{\mathbf{u}}[n]) = E\{\log(p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}))\} - E\left\{\log\left(\det\left(\tilde{\mathbf{J}}\right)\right)\right\} + \\ - \sum_{i=1}^{N} E\{\log(p_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}}))\}\end{aligned} \tag{7.33}$$

Given the free parameters $\mathbf{\Phi} = \left\{w_{ij}, \mathrm{Q}^{g}, \mathrm{Q}^{SC}\right\}$, eq. (7.33) leads to learning

rules for matrix weights w_{ij} and nonlinear compensating function $\mathbf{Q}_G^{NL}$ that are formally identical to those obtained (eqs. (7.30) and (7.32)) from the ME approach. In the learning rule for the matrix weights:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{r}^H \tag{7.34}$$

the term $\mathbf{\Psi} = \mathbf{\Psi}_R + j\mathbf{\Psi}_I$, where $\Psi_{Rk} = \dot{p}_{u_{Ri}}(u_{Ri})/p_{u_{Ri}}(u_{Ri})$ and $\Psi_{Ik} = \dot{p}_{u_{Ii}}(u_{Ii})/p_{u_{Ii}}(u_{Ii})$ contains the derivatives of the pdf of the i-th estimated source and are known as *score functions (SF)*[1].

The problem is now the estimation of these score functions: for this reason the block "Score Function Estimation" is inserted in the network in Figure 7.8.

In the totally blind case there is no a priori information on the hidden sources or output pdf. This is the reason why the output pdf should be estimated only during the learning phase, since output signals may change. Using a predetermined SF based on some a priori estimation is theoretically possible but leads to worse convergence performance. As a matter of fact, the matching between signals' pdf and the corresponding SFs is a critical issue for the learning algorithm, since it determines the performance in separation. In [183] the Gram-Charlier approximation was compared to the MLP estimator in estimating the pdf and the SFs. In [67] a polynomial function with adaptively learning coefficients was proposed. In [146, 40] a linear parametric estimation model based on a projection in a subspace spanned by nonlinear functions was described. All these approaches are limited by the fact that learning is not local and in several cases is performed off-line.

In [40] direct estimation of SFs by the *least mean square (LMS)* algorithm [75] was described. Parameters were estimated by minimizing the mean square error ε_k for each output channel k

$$\varepsilon_k = \frac{1}{2}E\left\{\left[\tilde{\Psi}_k\left(u_k, \mathbf{\Phi}\right) - \dot{p}_{u_k}\left(u_k\right)/p_{u_k}\left(u_k\right)\right]^2\right\}, \quad k = 1, \ldots, N \tag{7.35}$$

In eq. (7.35) $\tilde{\Psi}_k(u_k, \mathbf{\Phi})$ is the spline model of the SF, while $E\{\bullet\}$ is the expectation operator.

Derivation of the cost function as shown in Appendix C.4 yields the gradient expression for the spline control points Q^{SC}:

$$\begin{array}{l} \frac{\partial\varepsilon}{\partial Q_R^{SC}} = \mathbf{T_R}\mathbf{M}\mathbf{T_R}\mathbf{M}\mathbf{Q}_R^{SC} - \frac{1}{\Delta}\dot{\mathbf{T}}_R\mathbf{M} \\ \frac{\partial\varepsilon}{\partial Q_I^{SC}} = \mathbf{T_I}\mathbf{M}\mathbf{T_I}\mathbf{M}\mathbf{Q}_I^{SC} - \frac{1}{\Delta}\dot{\mathbf{T}}_I\mathbf{M} \end{array} \tag{7.36}$$

where Δ is the difference between the abscissas of adjacent control points.

[1] Sometimes score functions are defined in literature with minus sign.

7.5 The Renyi's Entropy

An analogue algorithm can be obtained considering the Renyi's Entropy of order α which is denoted by H_{R_α} [217, 158]:

$$H_{R_\alpha} = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{+\infty} \left(p_{\mathbf{y}} \left(\mathbf{y} \right) \right)^{\alpha} d\mathbf{y} \right) \tag{7.37}$$

It is useful to remember that $\lim_{\alpha \to 1} H_{R_\alpha} \left(\mathbf{y} \right) = H \left(\mathbf{y} \right)$, that is the classical Shannon entropy [55, 77].

Applying the eq. (7.37) to the joint network output $p_{\mathbf{y}}(\mathbf{y})$ and following the calculation proposed in the previous sections (see Appendix C.5), we obtain:

$$\Delta \mathbf{W} \propto \frac{\partial H_{R_\alpha}}{\partial \mathbf{W}} = -\frac{\alpha}{1-\alpha} \left[\mathbf{W}^{-H} + \mathbf{\Psi} \mathbf{x}^{H} \right] \tag{7.38}$$

We can see that eq. (7.38) is formally identical to eq. (7.17) or (7.30): the only thing that changes is a constant term which can be adsorbed in the learning rate.

Some authors used the Renyi's entropy of order $\alpha = 2$, called *quadratic entropy* [152, 153, 77]. The use of Renyi's Mutual Information was also proposed [79].

7.6 Other approaches

Before showing experimental results we want to summarize briefly other approaches to the problem of BSS in complex domain. The aim of this section is not to describe each approach (see references for this scope) but only to have an overview of the most meaningful results showing the progress of the research in this field. The existent approaches perform separation only in linear and instantaneous environment.

In 2000 a complex-valued version of the well-known *Fast ICA* algorithm algorithm was proposed [21]. This algorithm is based on a non-Gaussianity maximization derived from the Negentropy [55] but it works well only for circular sources [148, 134].

Recently in [113] a class of complex-valued ICA algorithms by maximizing the kurtosis cost function is derived.

Moreover Cardoso and Adali in [38] proposed an approach based on the maximization of the log-likelihood function extended in [114]. This approach is strongly linked with the INFOMAX principle [111]. A *maximum likelihood (ML)* solution to BSS problem was first derived in [146].

Lately Novey and Adali in [137] solved the limit on circularity in the complex Fast ICA using complex analytic functions by introducing the *complex maximization of non-Gaussianity (CMN)* algorithm. The authors also

showed the connection among ICA methods through maximization of non-Gaussianity, mutual information and maximum likelihood (ML) for the complex case.

7.7 Natural gradient adaptation

The stochastic gradient technique is widely used in literature for maximizing or minimizing a cost function. It is a very simple approach but, if data lies in a Riemannian space $\mathcal{S} = \{\vartheta \in \mathbb{C}\}$, the convergence can be very slow or the solution can fall in a local solution.

To avoid these problems the *natural gradient* is introduced [6, 77] for an arbitrary functional $J(\vartheta)$ defined in the space $\mathcal{S}$:

$$\tilde{\nabla} J(\vartheta) = G^{-1}(\vartheta) \nabla J(\vartheta) \tag{7.39}$$

where $\nabla J(\vartheta)$ is the standard stochastic gradient, $G^{-1}(\vartheta)$ is the inverse of the metric tensor and $\tilde{\nabla} J(\vartheta)$ is the natural gradient.

Amari has demonstrated in [6] that the inverse of the metric tensor in (7.39) is a very simple expression in the Lie group, i.e. the space of invertible matrices $\mathbf{W}_{\mathbf{n}\times\mathbf{n}}$, namely $GL(n, \mathbb{R})$. It is simply

$$\tilde{\nabla} J(W) = \nabla J(W) W^T W \tag{7.40}$$

Using (7.40) the ME algorithm (7.17) becomes the following one:

$$\Delta \mathbf{W} = (\mathbf{I} + \mathbf{\Psi u}^H)\mathbf{W} \tag{7.41}$$

This algorithm is numerically more efficient than the classical one since that it avoids the inversion of the $\mathbf{W}$ matrix.

In addition some new riemannian metrics are introduced in [10] and [180], in order to improve the convergence speed. The authors introduced four additional and alternative algorithms in a riemannian space. The new idea is based on an alternative natural gradient formulation, imposing that the inner product in a riemannian space be invariant under translation in such space. In this way five different natural gradient expression can be derived:

$$\tilde{\nabla}_R J(W) = \nabla J(W) W^T W \tag{7.42}$$

$$\tilde{\nabla}_L J(W) = W W^T \nabla J(W) \tag{7.43}$$

$$\tilde{\nabla}_{LR} J(W) = (W W^T) \nabla J(W) (W^T W) \tag{7.44}$$

$$\tilde{\nabla}_{RR} J(W) = \nabla J(W) (W^T W^T W W) \tag{7.45}$$

$$\tilde{\nabla}_{LL} J(W) = (W W W^T W^T) \nabla J(W) \tag{7.46}$$

namely the *right* natural gradient (7.42), which is the standard natural gradient (7.40) introduced in [6], the *left* natural gradient (7.43), the *right/left*

natural gradient (7.44), the *right/right* natural gradient (7.45) and the *left/left* natural gradient (7.46) rispectively.

Using (7.43)-(7.46) the ME algorithm (7.17) becomes [167]:

$$\Delta \mathbf{W} = \mathbf{W}(\mathbf{I} + \mathbf{W}^H \mathbf{\Psi}\mathbf{x}^H) \tag{7.47}$$

$$\Delta \mathbf{W} = \mathbf{W}\mathbf{W}^H(\mathbf{I} + \mathbf{\Psi}\mathbf{u}^H)\mathbf{W} \tag{7.48}$$

$$\Delta \mathbf{W} = (\mathbf{I} + \mathbf{\Psi}\mathbf{u}^H)\mathbf{W}^H\mathbf{W}\mathbf{W} \tag{7.49}$$

$$\Delta \mathbf{W} = \mathbf{W}\mathbf{W}\mathbf{W}^H(\mathbf{I} + \mathbf{W}^H \mathbf{\Psi}\mathbf{x}^H) \tag{7.50}$$

It should be noted that algorithms (7.48)-(7.50) do not satisfy the *equivariance property* [35].

7.7.1 The equivariance property

When a transformation on the data is equivalent to a transformation of the parameter, the notion of *equivarihnce* is of relevance. An estimator behaves "*equivariantly*" if it produces estimates that, under data transformation, are transformed accordingly.

Definition 25 *An estimator $\mathcal{A}$ for $\mathbf{A}$ ($\hat{\mathbf{A}} = \mathcal{A}(\mathbf{X})$) is said to be equivariant if for every non-singular matrix $\mathbf{M}$ it satisfies:*

$$\mathcal{A}(\mathbf{M}\mathbf{X}) = \mathbf{M}\mathcal{A}(\mathbf{X}) \tag{7.51}$$

The key property shared by equivariant estimators for source separation is that they offer *uniform performance*. This is to be understood in the following sense. Assume that the source signals are estimated as $\hat{\mathbf{s}}(t) = \hat{\mathbf{A}}^{-1}\mathbf{x}(t)$, where $\hat{\mathbf{A}}$ is obtained from an equivariant estimator. Then,

$$\begin{aligned} \hat{\mathbf{s}}(t) = (\mathcal{A}(\mathbf{X}))^{-1}\mathbf{x}(t) = (\mathcal{A}(\mathbf{A}\mathbf{S}))^{-1}\mathbf{A}\mathbf{s}(t) = \\ = (\mathbf{A}\mathcal{A}(\mathbf{S}))^{-1}\mathbf{A}\mathbf{s}(t) = \mathcal{A}(\mathbf{S})^{-1}\mathbf{s}(t) \end{aligned} \tag{7.52}$$

where we have only used the equivariance property in eq. (7.51).

The last equality reveals that source signals estimated by an equivariant estimator $\mathcal{A}$ for a particular realization $\mathbf{S}$ depends only on $\mathbf{S}$ but do not depend on the mixing matrix $\mathbf{A}$. It follows that in terms of signal separation, the performance of an equivariant algorithm does not depend at all on the mixing matrix.

This fact implies that, posing $\mathbf{C} = \mathbf{W}\mathbf{A}$, the natural gradient algorithm (7.41) depends only on $\mathbf{C}$:

$$\Delta \mathbf{W} \cdot \mathbf{A} = (\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{W}\mathbf{A} = (\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{C} \tag{7.53}$$

It is clear that the expression of $\Delta\mathbf{W}$ depends only on $\mathbf{C}$.

This property is not satisfied by all the other algorithms (7.47)-(7.50). Let we suppose that the mixing matrix $\mathbf{A}$ is a unitary matrix ($\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A} = \mathbf{I}$), then:

1. $\Delta\mathbf{W} \cdot \mathbf{A} = \mathbf{W}(\mathbf{I} + \Psi\mathbf{x}^H)\mathbf{A}$;
2. $\Delta\mathbf{W} \cdot \mathbf{A} = \mathbf{W}\mathbf{W}^H(\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{W}\mathbf{A} = \mathbf{W}\mathbf{A}\mathbf{A}^H\mathbf{W}^H(\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{C} = \mathbf{C}\mathbf{C}^H(\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{C}$;
3. $\Delta\mathbf{W} \cdot \mathbf{A} = (\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{W}^H\mathbf{W}\mathbf{W}\mathbf{A} = (\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{W}^H\mathbf{A}^H\mathbf{A}\mathbf{W}\mathbf{C} = (\mathbf{I} + \Psi\mathbf{u}^H)\mathbf{C}^H\mathbf{C}\mathbf{C}$;
4. $\Delta\mathbf{W} \cdot \mathbf{A} = \mathbf{W}\mathbf{W}\mathbf{W}^H(\mathbf{I} + \Psi\mathbf{x}^H)\mathbf{A}$.

We note that the **left** and **left/left** algorithms in eqs. (7.47) and (7.50) respectively do not satisfy the equivariance property: we expect a poor performance behavior (see the experimental tests in Section 9.4).

8

Special Topics

—Gather up the fragments that remain, that nothing be lost.
John 6:12

In this chapter three interesting results are shown. In particular we show that splines are universal functions for the blind source separation problem: the *flexibility* of this kind of function makes splines suitable for separation of both super-gaussian and sub-gaussian sources. A second result is that the splitting solution allows the proposed algorithm to recover the original phase. Moreover it shows a Cramér-Rao lower bound for ICA solution.

8.1 The universality of spline function

A main issue in Blind Signal Separation (BSS) problem is the *stability analysis* of the algorithm [7, 215, 125] and the properties which have to be satisfied by the nonlinear function involved in the learning process.

Several studies on the *convergence* of the algorithm (7.41) exist and it is pointed out that the function $\mathbf{\Psi}(\mathbf{u})$ must satisfy the following equation [7, 215, 125]

$$E\left\{\mathbf{\Psi}^{'}(\mathbf{u})\right\} E\left\{\mathbf{u^2}\right\} + E\left\{\mathbf{\Psi}(\mathbf{u})\mathbf{u^T}\right\} > 0 \tag{8.1}$$

In order to demonstrate that spline function satisfy the eq. (8.1) it is useful to introduce some results on *score functions* (*SC*) derived from [11].

Definition 26 (Score Function) *The score function of a scalar random variable*

x is the log derivative of its density[1]*:*

$$\varphi_x(x) \triangleq \frac{d}{dx} \ln p_x(x) = \frac{p'_x(x)}{p_x(x)} \tag{8.2}$$

In conjunction with this definition, we define two different types of score functions for a random vector $\mathbf{x} = [x_1, \ldots, x_N]^T$:

Definition 27 (MSF) *The **Marginal Score Function (MSF)** of $\mathbf{x}$ is the vector of score functions of its components. That is:*

$$\phi_{\mathbf{x}}(\mathbf{x}) \triangleq [\phi_1(x_1), \ldots, \phi_N(x_N)]^T \tag{8.3}$$

where

$$\phi_i(x_i) \triangleq \frac{d}{dx_i} \ln p_{x_i}(x_i) = \frac{p'_{x_i}(x_i)}{p_{x_i}(x_i)}$$

Definition 28 (JSF) *The **Joint Score Function (JSF)** of $\mathbf{x}$ is the gradient of $\ln p_{\mathbf{x}}(\mathbf{x})$, that is:*

$$\varphi_{\mathbf{x}}(\mathbf{x}) \triangleq [\varphi_1(x_1), \ldots, \varphi_N(x_N)]^T \tag{8.4}$$

where

$$\varphi_i(x_i) \triangleq \frac{\partial}{\partial x_i} \ln p_{\mathbf{x}}(\mathbf{x}) = \frac{\frac{\partial}{\partial x_i} p_{\mathbf{x}}(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})}$$

Lemma 7 *The components of a random vector $\mathbf{x} = [x_1, \ldots, x_N]^T$ are independent if and only if*

$$\varphi_{\mathbf{x}}(\mathbf{x}) = \phi_{\mathbf{x}}(\mathbf{x})$$

Lemma 8 *Let x be a random variable with the PDF $p_x(x)$ and the score function $\varphi_x(x)$. Moreover, let f be a continuously differentiable function and $\lim_{x \to \pm\infty} f(x) p_x(x) = 0$. Then*

$$E\{f(x) \varphi_x(x)\} = E\{f'(x)\} \tag{8.5}$$

Corollary 2 *For a bounded random variable x, we have:*

$$E\{\varphi_x(x) x\} = 1 \tag{8.6}$$

Therefore it is helpful to give the following

[1] Sometimes score functions are defined in literature with minus sign.

$$\varphi_x(x) \triangleq -\frac{d}{dx} \ln p_x(x) = -\frac{p'_x(x)}{p_x(x)}$$

Proposition 5 *Let* $\mathbf{u} = \mathbf{A}\mathbf{x}$*, where* $\mathbf{x}$ *and* $\mathbf{u}$ *are random vectors and* $\mathbf{W}$ *is a non-singular square matrix. Then*

$$\varphi_{\mathbf{u}}(\mathbf{u}) = \mathbf{W}^{-T}\varphi_{\mathbf{x}}(\mathbf{x}) \tag{8.7}$$

Proof. From $\mathbf{u} = \mathbf{A}\mathbf{x}$ we have

$$p_{\mathbf{u}}(\mathbf{u}) = \frac{p_{\mathbf{x}}(\mathbf{x})}{|det\mathbf{W}|} \Rightarrow lnp_{\mathbf{x}}(\mathbf{x}) = lnp_{\mathbf{u}}(\mathbf{u}) + ln\,|det\mathbf{W}|$$

Therefore, for $i = 1, \ldots, N$ we ca write

$$\begin{aligned}\varphi_{\mathbf{x},i}(\mathbf{x}) &= -\frac{\partial}{\partial x_i}\ln p_{\mathbf{x}}(\mathbf{x}) = -\frac{\partial}{\partial x_i}\ln p_{\mathbf{u}}(\mathbf{u}) = \\ &= -\sum_{k=1}^{N}\frac{\partial}{\partial u_k}\ln p_{\mathbf{u}}(\mathbf{u})\cdot\frac{\partial u_k}{\partial x_i} = \sum_{k=1}^{N} w_{ki}\varphi_{\mathbf{u},k}(\mathbf{u})\end{aligned}$$

where $\varphi_{\mathbf{x},i}(\mathbf{x})$ and $\varphi_{\mathbf{u},i}(\mathbf{u})$ denote the i-th components of the joint score function of $\mathbf{x}$ and $\mathbf{y}$, respectively. From the above relation, we have $\varphi_{\mathbf{u}}(\mathbf{u}) = \mathbf{W}^{-T}\varphi_{\mathbf{x}}(\mathbf{x})$, which proves the property. ■

In Section 7 we have seen that the flexible activation functions $h_i(x_i)$ adapt their shape to the shape of the cumulative density function (cdf) of the original sources [162]. In this way it is evident that for the real-valued case the terms Ψ_i of $\mathbf{\Psi}$ vector in the learning rule, coincides with the score functions. In fact

$$\Psi_i = \frac{h_i''}{h_i'} = \frac{p_{x_i}'}{p_{x_i}} \equiv \varphi_i$$

This fact justifies the following

Theorem 18 *The spline function previous introduced always satisfies the condition* (8.1).

Proof. It is possible to explicitly quantify the single terms in eq. (8.1). From eqs. (8.6) and (8.7) we obtain

$$\begin{aligned}&E\left\{\mathbf{\Psi}(\mathbf{u})\,\mathbf{u}^T\right\} = E\left\{\mathbf{W}^{-T}\mathbf{\Psi}(\mathbf{x})\,\mathbf{x}^T\mathbf{W}^T\right\} = \\ &= \mathbf{W}^{-T}E\left\{\mathbf{\Psi}(\mathbf{x})\,\mathbf{x}^T\right\}\mathbf{W}^T = \mathbf{W}^{-T}\mathbf{W}^T = \mathbf{I}\end{aligned}$$

and therefor for each component $E\{\Psi_i(u_i)\,u_i\} = 1, \forall i = 1, \ldots, N$. Moreover

$$E\left\{u^2\right\} = \sigma_u^2 \geq 0$$

From eq. (8.5) we can obtain

$$E\left\{\mathbf{\Psi}'(u)\right\} = E\left\{\mathbf{\Psi}^2(u)\right\} \geq 0$$

This result justifies eq. (8.1), hence the theorem. ■

This result allows us to consider splines as the universal functions for the Blind Source Separation problem: in fact the *flexibility* of this kind of functions makes splines suitable for separation of both super-gaussian and sub-gaussian sources [162]. In this way the constraint in eq. (8.1) is alway satisfied, as stated by Theorem 18.

Performing some simple mathematical manipulation is not too difficult to demonstrate Theorem 18 in the complex domain. In this case the terms Ψ_{Ri} and Ψ_{Ii} of the complex vector $\mathbf{\Psi}$ are a little more complicated than the real-valued case.

8.2 Phase recovery

As we have seen in Chapter 4 the *scaling ambiguity* in the complex domain causes a *rotation ambiguity* too. In this sense it is not possible to recover the original phase of the original source generally. In this section we introduce two constraints that allows the proposed algorithms to recover the phase information [78].

In conventional BSS using an algorithm such as the complex INFOMAX algorithm, the recovered complex source signals may have undetermined component order and phase. While component order may be recoverable using some side information in practice (e.g., sources with different signal constellations or some frame-level information), phase rotation is undesirable in communications signal. In such applications, we can usually assume that the complex signals have independent I/Q components [17, 18, 205, 204, 213]. Note that this assumption holds for equally likely points in square *quadrature amplitude modulation* (*QAM*) constellations but not for *phase shift keying* (*PSK*) constellations. If channel coding is involved, strict I/Q independence may not hold, although approximate I/Q independence may be a reasonable assumption. To recover the phase rotation of each source at the time of source separation, a *constrained* BSS technique can be defined to separate sources as well as recover source phases for I/Q-independent sources. The basic idea of this constrained I/Q BSS algorithm is as follows: since the I/Q components of each source are statistically independent, they are also independent of I/Q components of other sources, and the original complex source mixtures can be considered as $2n$ real source mixtures. These $2n$ real source mixtures are composed of mutually independent I/Q parts of the n independent complex sources. Source separation techniques can then be applied to separate the $2n$ independent real signals. Because of the order indeterminacy of BSS methods, special constraints are required so that the separated signals retain correct I/Q association [78].

Without loss of generality, consider the case of two independent complex sources with independent I/Q components. The two complex source

mixtures can be considered as four real mixtures of the I/Q parts of the two complex sources.

Let we pose $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$ e $\mathbf{x}_i = [x_{Ri}, x_{Ii}]^T$, and similarly $\mathbf{s} = [\mathbf{s}_1, \ldots, \mathbf{s}_N]^T$ e $\mathbf{s}_i = [s_{Ri}, s_{Ii}]^T$. The BSS model can be expressed as

$$\left[\begin{array}{c} x_{1R} \\ x_{1I} \\ \hline x_{2R} \\ x_{2I} \\ \hline \vdots \\ \hline x_{NR} \\ x_{NI} \end{array}\right] = \left[\begin{array}{cc|cc|c|cc} a_{11R} & -a_{11I} & a_{12R} & -a_{21I} & \cdots & a_{1NR} & -a_{1NI} \\ a_{11I} & a_{11R} & a_{12I} & a_{12R} & \cdots & a_{1NI} & a_{1NR} \\ \hline a_{21R} & -a_{21I} & a_{22R} & -a_{22I} & \cdots & a_{2NR} & -a_{2NI} \\ a_{21I} & a_{21R} & a_{22I} & a_{22R} & \cdots & a_{2NI} & a_{2NR} \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline a_{N1R} & -a_{N1I} & a_{N2R} & -a_{N2I} & \cdots & a_{NNR} & -a_{NNI} \\ a_{N1I} & a_{N1R} & a_{N2I} & a_{N2R} & \cdots & a_{NNI} & a_{NNR} \end{array}\right] \cdot \left[\begin{array}{c} s_{1R} \\ s_{1I} \\ \hline s_{2R} \\ s_{2I} \\ \hline \vdots \\ \hline s_{NR} \\ s_{NI} \end{array}\right] \tag{8.8}$$

Eq. (8.8) can be rewritten in a more compact form as

$$\left[\begin{array}{c} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \\ \hline \vdots \\ \hline \mathbf{x}_N \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{A_{11}} & \mathbf{A_{12}} & \cdots & \mathbf{A_{1N}} \\ \hline \mathbf{A_{21}} & \mathbf{A_{22}} & \cdots & \mathbf{A_{2N}} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{A_{N1}} & \mathbf{A_{N2}} & \cdots & \mathbf{A_{NN}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{s}_1 \\ \hline \mathbf{s}_2 \\ \hline \vdots \\ \hline \mathbf{s}_N \end{array}\right] \tag{8.9}$$

where $\mathbf{A_{ij}} = \begin{bmatrix} a_{ijR} & -a_{ijI} \\ a_{ijI} & a_{ijR} \end{bmatrix}$ and $\det(\mathbf{A_{ij}}) = a_{ijR}^2 + a_{ijI}^2 = |a_{ij}|^2$.

We assume a similar structure for the de-mixing model

$$\left[\begin{array}{c} y_{1R} \\ y_{1I} \\ \hline y_{2R} \\ y_{2I} \\ \hline \vdots \\ \hline y_{NR} \\ y_{NI} \end{array}\right] = \left[\begin{array}{cc|cc|c|cc} w_{11R} & -w_{11I} & w_{12R} & -w_{21I} & \cdots & w_{1NR} & -w_{1NI} \\ w_{11I} & w_{11R} & w_{12I} & w_{12R} & \cdots & w_{1NI} & w_{1NR} \\ \hline w_{21R} & -w_{21I} & w_{22R} & -w_{22I} & \cdots & w_{2NR} & -w_{2NI} \\ w_{21I} & w_{21R} & w_{22I} & w_{22R} & \cdots & w_{2NI} & w_{2NR} \\ \hline \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \hline w_{N1R} & -w_{N1I} & w_{N2R} & -w_{N2I} & \cdots & w_{NNR} & -w_{NNI} \\ w_{N1I} & w_{N1R} & w_{N2I} & w_{N2R} & \cdots & w_{NNI} & w_{NNR} \end{array}\right] \cdot \left[\begin{array}{c} x_{1R} \\ x_{1I} \\ \hline x_{2R} \\ x_{2I} \\ \hline \vdots \\ \hline x_{NR} \\ x_{NI} \end{array}\right] \tag{8.10}$$

Eq. (8.10) can be rewritten in a more compact form as

$$\left[\begin{array}{c} \mathbf{y}_1 \\ \hline \mathbf{y}_2 \\ \hline \vdots \\ \hline \mathbf{y}_N \end{array}\right] = \left[\begin{array}{c|c|c|c} \mathbf{W_{11}} & \mathbf{W_{12}} & \cdots & \mathbf{W_{1N}} \\ \hline \mathbf{W_{21}} & \mathbf{W_{22}} & \cdots & \mathbf{W_{2N}} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{W_{N1}} & \mathbf{W_{N2}} & \cdots & \mathbf{W_{NN}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \\ \hline \vdots \\ \hline \mathbf{x}_N \end{array}\right] \tag{8.11}$$

where $\mathbf{W_{ij}} = \begin{bmatrix} w_{ijR} & -w_{ijI} \\ w_{ijI} & w_{ijR} \end{bmatrix}$ and $\det(\mathbf{W_{ij}}) = w_{ijR}^2 + w_{ijI}^2 = |w_{ij}|^2$.

By imposing the constraint given in (8.10) and (8.11) on the structure of the real separating matrix $\mathbf{W}_k$ at each iteration k of an adaptive BSS

algorithm seeking the real independent source signals, the correct I/Q association can be obtained.

At each iteration of an unconstrained adaptive BSS algorithm, an updated real separating matrix is first obtained, in which each sub-matrix block is denoted as $\mathbf{W}_{ij}^u$ with the entries w_{ijk}^u ($k = 1, 2, 3, 4$). To enforce the constraint of (8.11), we impose the modification in the following lemma 9 for each sub-matrix $\mathbf{W}_{ij}^u$. The resulting constrained separation matrix is used in the next iteration.

Lemma 9 *The model* (8.11) *allows the separation of I/Q sources under the following constraint*

$$\mathbf{W_{ij}} = \frac{1}{2}\left\{\mathbf{W}_{ij}^u + \det\left(\mathbf{W}_{ij}^u\right) \cdot (\mathbf{W}_{ij}^u)^{-T}\right\} \tag{8.12}$$

where $(\mathbf{W}_{ij}^u)^{-T}$ *denotes* $[(\mathbf{W}_{ij}^u)^T]^{-1}$.

Proof. Eq. (8.12) is derived from the particular structure of (8.9) and (8.11). In fact every sub-matrix block has the same structure $\begin{bmatrix} w_{ijR} & -w_{ijI} \\ w_{ijI} & w_{ijR} \end{bmatrix}$. A generic matrix $\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$ could be transformed in the desired one by the following summation

$$\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} + \begin{bmatrix} \delta & -\gamma \\ -\beta & \alpha \end{bmatrix}.$$

But we have

$$\begin{bmatrix} \delta & -\gamma \\ -\beta & \alpha \end{bmatrix} = Agg\left(\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}^T\right) = \det\left(\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}\right) \cdot inv\left(\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}^T\right),$$

from that eq. (8.12) can be directly derived by a division for 2. ■

It can be shown that by using the constrained I/Q BSS, we can maintain the correct I/Q association for each I/Q-independent source, with only $\pi/2$-phase ambiguity possible during the whole process of simultaneous source separation and phase recovery. The reason for the remaining $\pi/2$-phase ambiguity is that a sign indeterminacy remains for the separated source components. The $\pi/2$ ambiguity can be handled by differential encoding in communications.

As discussed above, constrained I/Q BSS in the complex INFOMAX algorithm can separate sources and recover source phases. Alternatively, we can apply the I/Q independence property implicitly in the choice of the component scalar complex-valued nonlinear function h in the complex INFOMAX algorithm to achieve source phase recovery. We call this the *constrained nonlinear function method*. In this method, the scalar complex-valued nonlinear function for the complex INFOMAX algorithm is constrained to have the decomposition stated in the following

Theorem 19 *The de-mixing model obtained by eq.* (8.12) *is equivalent to the complex model where the nonlinearity* $h(z)$ *(the activation function) is of the following type*

$$h(z) = h^R(z_R) + jh^I(z_I) \tag{8.13}$$

In other words we adopt the splitting function in eqs. (7.12) *and* (5.3).

Proof. It is possible to show that the model (8.13) is equivalent to the model (8.12). For semplicity we consider the trivial case of only one complex source $\mathbf{x} = \mathbf{A}\mathbf{s}$ obtaining $w^{k+1} = w^k - \mu\left\{[1 + hy^*]\, w^k\right\}$, where $h = h_R + jh_I$. Considering the model (8.12) we obtain

$$\begin{gathered} w^{k+1} = w_R^{k+1} + jw_I^{k+1} = \left(w_R^k + jw_I^k\right) + \\ -\mu\left\{[1 + (h_R + jh_I)(y_R - jy_I)]\left(w_R^k + jw_I^k\right)\right\} = \\ = w_R^k - \mu\left[(1 + h_R y_R + h_I y_I)\, w_R^k + (h_R y_I - h_I y_R)\, w_I^k\right] + \\ +j\left\{w_I^k - \mu\left[(1 + h_R y_R + h_I y_I)\, w_I^k + (h_I y_R - h_R y_I)\, w_R^k\right]\right\} \end{gathered}$$

and considering the real and imaginary part respectively:

$$\begin{aligned} w_R^{k+1} &= w_R^k - \mu\left[(1 + h_R y_R + h_I y_I)\, w_R^k + (h_R y_I - h_I y_R)\, w_I^k\right] \\ w_I^{k+1} &= w_I^k - \mu\left[(1 + h_R y_R + h_I y_I)\, w_I^k + (h_I y_R - h_R y_I)\, w_R^k\right] \end{aligned} \tag{8.14}$$

Considering the model (8.13) instead, we obtain:

$$\begin{gathered} \begin{bmatrix} w_{11}^{k+1} & w_{12}^{k+1} \\ w_{21}^{k+1} & w_{22}^{k+1} \end{bmatrix} = \begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix} + \\ -\mu\left\{\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} h_R \\ h_I \end{bmatrix}\begin{bmatrix} y_R & -y_I \end{bmatrix}\right)\begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix}\right\} = \\ = \begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix} + \\ -\mu\left\{\begin{bmatrix} 1 + h_R y_R & -h_R y_I \\ h_I y_R & 1 - h_I y_I \end{bmatrix}\begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix}\right\} = \\ = \begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix} + \\ -\mu\begin{bmatrix} (1 + h_R y_R)\, w_{11}^k - h_R y_I w_{21}^k & (1 + h_R y_R)\, w_{12}^k - h_R y_I w_{22}^k \\ h_I y_R w_{11}^k + (1 - h_I y_I)\, w_{21}^k & h_I y_R w_{12}^k + (1 - h_I y_I)\, w_{22}^k \end{bmatrix} \end{gathered}$$

using the constraint in eq. (8.13), it is possible to recover the coefficients:

$$
\begin{aligned}
w_R^{k+1} &= \frac{1}{2}\left(w_{11}^{k+1} + w_{22}^{k+1}\right) = \\
&= \frac{1}{2}\left\{\left(w_{11}^{k} + w_{22}^{k}\right) - \mu\left[\left(2 + h_R y_R - h_I y_I\right) w_R^k - \left(h_R y_I - h_I y_R\right) w_I^k\right]\right\} = \\
&= w_R^k - \mu\left\{\left(1 + \tfrac{h_R y_R + h_I y_I}{2}\right) w_R^k - \left(\tfrac{h_R y_I - h_I y_R}{2}\right) w_I^k\right\} \\
w_I^{k+1} &= \frac{1}{2}\left(w_{21}^{k+1} - w_{12}^{k+1}\right) = \\
&= \frac{1}{2}\left\{\left(w_{21}^{k} - w_{12}^{k}\right) - \mu\left[\left(h_R y_I + h_I y_R\right) w_R^k + \left(2 + h_R y_R - h_I y_I\right) w_I^k\right]\right\} = \\
&= w_I^k - \mu\left\{\left(\tfrac{h_R y_I - h_I y_R}{2}\right) w_R^k + \left(1 + \tfrac{h_R y_R + h_I y_I}{2}\right) w_I^k\right\}
\end{aligned}
\tag{8.15}
$$

It can be seen that eq. (8.15) is formally identical to eq. (8.14) for less than some scaling factors. This fact demonstrates the Theorem 19. ■

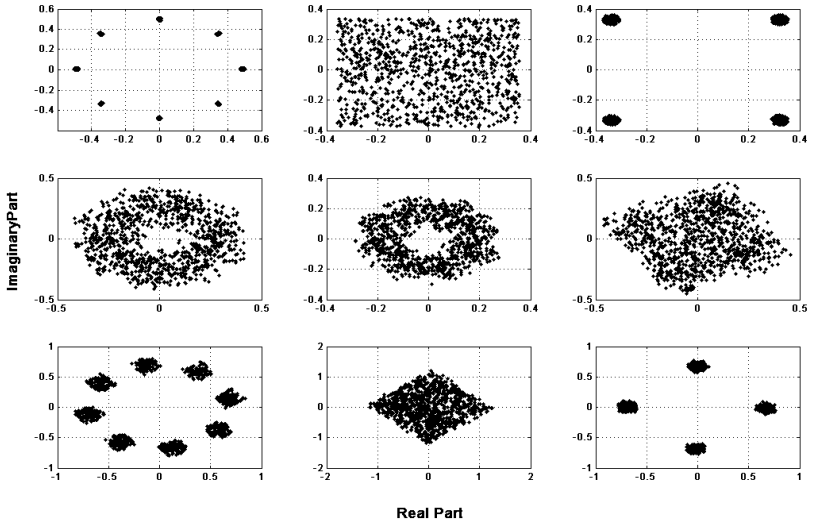

Fig. 8.1: Scatter plot of signals using a complex $tanh(z)$ activation function

The constrained nonlinear function approach does not require doubling of the dimension (from complex to real) and does not impose a separate constraint on the $\mathbf{W}$ updates.

The same consideration can be done on another blind source separation algorithm: the *EASI algorithm* [35] with similar results.

Figure 8.1 shows an example of separation of a 8-PSK signal, a 4-QAM signal and a uniform noise utilizing a complex $tanh(z)$ activation function as described in [2] and not a splitting solution. It is evident from the figure the the phase is not recovered: in fact the scatter plot of the recovered signal (third row) is rotated with respect the original signal (first row).

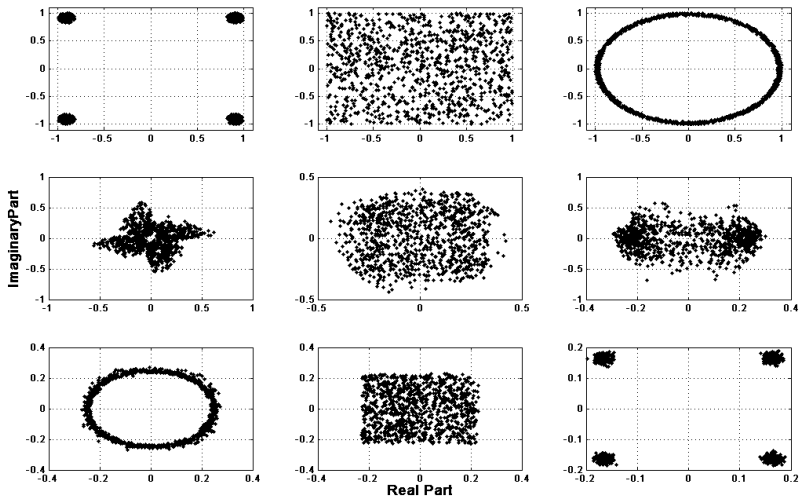

Fig. 8.2: Scatter plot of signals using a splitting activation function

Figure 8.2 shows an example of separation of a 8-PSK signal, a 4-QAM signal and a uniform noise utilizing a splitting activation function as described in [200, 165, 212]. It is evident from the figure the the phase is now recovered: in fact the scatter plot of the recovered signal (third row) is not rotated with respect the original signal (first row).

Remain a doubt on the validity of the Theorem 19 for strongly non-I/Q signal. For demonstrate the validity of this approach we perform separation of a *Bernoulli's lemniscate*, which is an artificial signal, but has the real and imaginary part strongly correlated. Figure 8.3 shows the effectiveness of the phase recovery in this case, too.

8.3 The Cramér-Rao lower bound for ICA

It is highly useful to have a *lower bound* for the statistical variability (*accuracy*) of an estimator. *Cramér-Rao bound* (*CRB*) provides a lower bound on the covariance matrix of any unbiased estimator of a parameter vector. CRB, which is the inverse of the *Fisher information matrix* (*FIM*), can be used e.g., to show that an unbiased estimator is *uniformly minimum variance unbiased* (*UMVU*) estimator. CRB is also related to asymptotic optimality theory.

Despite of the increased interest in ICA during the past two decades, a closed-form expression for the CRB for the de-mixing matrix estimation has been established very recently in [139, 140] and in [198, 197, 196]. CRB is derived indirectly in [33, 115, 145, 171, 220] via asymptotic approximations of

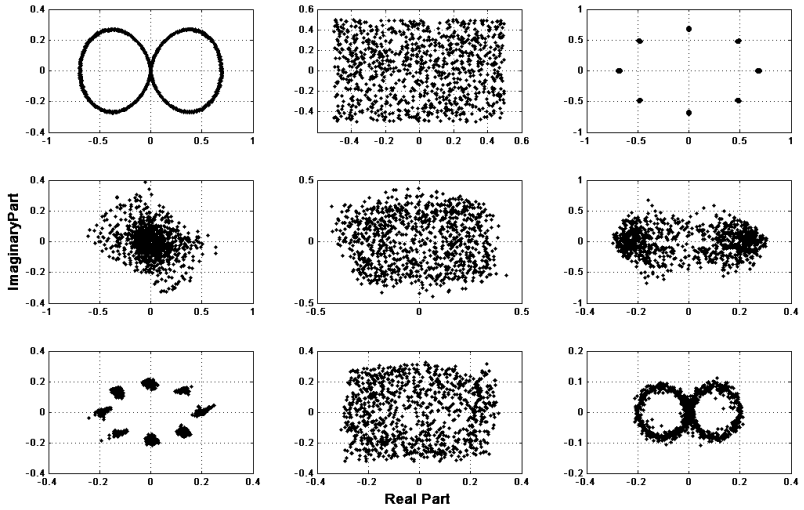

Fig. 8.3: Scatterplot of strongly non-I/Q signals using the splitting activation function

the likelihood or via asymptotic covariance matrix of the *maximum-likelihood* (*ML*) estimator of a transformed parameters such as the interference-o-signal ratio.

Suppose i.i.d. observations $x_1, \ldots, x_N$ are distributed as $\mathbf{x}$ having the pdf $p_\theta(\mathbf{x})$ with parameter vector $\theta \in \Theta$. The inverse of the FIM of θ

$$\mathcal{I}_\theta = E\left\{\nabla_\theta \ln p_\theta(\mathbf{x}) \left[\nabla_\theta \ln p_\theta(\mathbf{x})\right]^T\right\} \tag{8.16}$$

where ∇_θ is the gradient operator, gives, under regularity conditions, the CRB on the covariance matrix of an unbiased estimator $\hat{\theta}$ of θ in the sense that

$$\mathrm{cov}(\hat{\theta}) \geq n^{-1}\mathcal{I}_\theta^{-1} \tag{8.17}$$

Above, for symmetric matrices B and C, the notation $B \geq C$ implies that $B - C$ is positive semidefinite. The CRB (8.17) thus implies, for example, that $\mathrm{var}(\hat{\theta}_i) \geq n^{-1}(\mathcal{I}_\theta^{-1})_{ii}$, where $\hat{\theta}_i$ denotes the i-th component of $\hat{\theta}$ and $(\mathcal{I}_\theta^{-1})_{ii}$ the i-th element of $\mathcal{I}_\theta^{-1}$. CRB is also related to asymptotic optimality theory in the sense that asymptotic covariance matrix of the ML estimator coincides with $\mathcal{I}_\theta^{-1}$. Recall however that there may not exist an unbiased estimator that attains the CRB for all $\theta \in \Theta$.

Recall the scaling, sign and permutation ambiguity of the ICA problem: if $\mathbf{\Lambda}$ is a $n \times n$ diagonal matrix and $\mathbf{P}$ is a $n \times n$ permutation matrix, then $\mathbf{x} = (\mathbf{A}\mathbf{P}^{-1}\mathbf{\Lambda}^{-1})(\mathbf{\Lambda}\mathbf{P}\mathbf{s})$, where $\mathbf{\Lambda}\mathbf{P}\mathbf{s}$ has independent components as well. Therefore, scales of s_i's can be fixed, e.g., by imposing $\mathrm{var}(s_i) = 1$, $i = 1, \ldots, N$. This scaling convention is common in ICA and it renders $\mathbf{A}$

(respectively, $\mathbf{W}$) unique up to permutation and sign of its columns (respectively, rows).

First we form the parameter vector

$$\theta = \text{vec}\left(\mathbf{W}^T\right) = \left[\mathbf{w}_1^T, \ldots, \mathbf{w}_n^T\right]^T \in \mathbb{R}^{n^2} \tag{8.18}$$

where $\mathbf{w}_i \in \mathbb{R}^n$ are the row vectors of and the "**vec**" is the well-known *vectorizing operator* [117], namely, if B is $n \times m$ matrix, then $\text{vec}(B)$ is a nm-dimensional vector formed by stacking the column vectors of the matrix B on top of each other. The pdf of $\mathbf{x} = \mathbf{As}$ is $p_\theta(\mathbf{x}) = |\det(\mathbf{W})| \prod_{i=1}^N p_i\left(\mathbf{w}_i^T\mathbf{x}\right)$, where p_i denotes the pdf of s_i. Use of matrix derivatives gives

$$\frac{\partial}{\partial \mathbf{W}^T} \ln p_\theta(\mathbf{x}) = \mathbf{A} - \mathbf{x}\varphi(\mathbf{Wx})^T \tag{8.19}$$

where $\varphi(\mathbf{s}) = (\varphi_1(s_1), \ldots, \varphi_n(s_n))^T$ and $\varphi_i(s_i) = -p_i'(s_i)/p_i(s_i)$ is the *score function* of the i-th IC. The *Fisher score* of the parameter (8.18) in the ICA model can now be calculated by

$$\nabla_\theta \ln p_\theta(\mathbf{x}) = \text{vec}\left\{\frac{\partial}{\partial \mathbf{W}^T} \ln p_\theta(\mathbf{x})\right\} \tag{8.20}$$

The following assumptions on i-th IC s_i for $i = 1, \ldots, N$ are made.

1. s_i has zero mean $E\{s_i\} = 0$ and unit variance $\text{var}(s_i) = E\left\{s_i^2\right\} = 1$ and only one of the IC's $s_1, \ldots, s_n$ can have a Gaussian distribution.
2. The pdf p_i of s_i satisfy:
   (a) p_i is continuous with contiguous support, $p_i(s) > 0$ and $p_i'(s) = (d/ds)p_i(s)$ exist $\forall s$ on the support of the density p_i;
   (b) $sp_i(s)$ tends to zero as s tends to the boundaries of the support of p_i.
3. The following variances:

$$\kappa_i = \text{var}\{\varphi_i(s_i)\} = E\left\{\varphi_i^2(s_i)\right\} = -\int \varphi_i(s) p_i'(s)\, ds \tag{8.21}$$

$$\lambda_i = \text{var}\{\varphi_i(s_i)\, s_i\} = E\left\{\varphi_i^2(s_i)\, s_i^2\right\} - 1 = -\int \varphi_i(s)\, p_i'(s) s^2 ds - 1 \tag{8.22}$$

exist and are finite.

The assumption of finite variance in 1) turns out to be crucial for the existence of the FIM. Such a restriction necessarily excludes, for instance, the Cauchy distribution which does not possess finite variance. The mean of s_i

is irrelevant and is, for ease of exposition, assumed to be zero. The necessity of at most one Gaussian component is a necessary restriction in ICA [50].

Assumption 2(a) is mainly needed for the existence of the Fisher score (8.20). Assumption 2(b) is not very restrictive and quite reasonable for densities with infinite support. 2(b) implicitly implies that $p_i(s)$ tends to zero as s tends to the boundaries of the support of p_i, which subsequently implies that $E\{\phi_i(s_i)\} = -\int p_i'(s)ds = 0$. Hence, 2(b) may not often be satisfied for densities with finite or semi-finite support. Clearly, e.g., the (zero mean) uniform distribution and the exponential distribution do not satisfy 2). Note that the zero mean Laplace distribution satisfies 2(b) but it does not satisfy 1(b) since it is not differentiable at $s = 0$. Nevertheless, Laplace distribution can be approximated to within arbitrary precision by a valid pdf that does satisfy 2).

For finiteness of the variances in (8.21) and (8.22), the respective integrands in (8.21) and (8.22), i.e., $l_i(s) = \varphi_i(s)p_i'(s)$ and $g_i(s) = \varphi_i(s)p_i'(s)s^2 = l_i(s)s^2$ need to decay rapidly enough to zero as s tends to $\pm\infty$ in case of infinite support sources, or, be bounded in case of finite support sources. For example, the zero mean Rayleigh distribution which is commonly used in communications theory satisfies assumptions 1) and 2), but not 3). It can be shown that $\kappa_i \geq 1$ with equality if and only if s_i is a Gaussian random variable and that $\lambda_i > 0$.

If $p_i''(s)$ (second derivative of p_i) exists at all s, then using the result in Lemma 8 (eq. 8.5), κ_i can be calculated by

$$\kappa_i = E\left\{\varphi_i'(s_i)\right\}$$

provided that

4. $p_i'(s) \to 0$ as s tends to the boundaries of the support of p_i.

Note that 4) is satisfied for all infinite support sources. Thus, the assumption 4) should be checked for distributions with finite or semi-finite support only. Similarly, if we assume that $p_i''(s)$ exists at all s, then

$$\lambda_i = E\left\{\varphi_i'(s_i)\, s_i^2\right\} + 1$$

providing that

5. $p_i'(s)s^2 \to 0$ as s tends to the boundaries of the support of p_i.

Note that 4) implies 5) if p_i has finite support, but not in the case of infinite or semi-finite support.

We may calculate the FIM (8.16) using the expression

$$\begin{aligned}\mathcal{I}_\theta = E\left\{\mathrm{vec}\left[\mathbf{A}\left(\mathbf{I} - \mathbf{s}\varphi(\mathbf{s})^T\right)\right]\mathrm{vec}\left[\mathbf{A}\left(\mathbf{I} - \mathbf{s}\varphi(\mathbf{s})^T\right)\right]^T\right\} = \\ = (\mathbf{I}\otimes\mathbf{A})\, E\left\{\mathrm{vec}\left[\mathbf{I} - \mathbf{s}\varphi(\mathbf{s})^T\right]\right\}\times \\ \times\mathrm{vec}\left[\mathbf{I} - \mathbf{s}\varphi(\mathbf{s})^T\right]^T\left(\mathbf{I}\otimes\mathbf{A}^T\right)\end{aligned} \tag{8.23}$$

where $\mathbf{I}$ denotes the identity matrix. Here we applied eq. (8.20) and algebraic properties involving the vec transformation and the Kronecker product and that $\mathbf{x}$ follows ICA model (2.1), i.e., $\mathbf{x} = \mathbf{A}\mathbf{s}$ and $\mathbf{W}\mathbf{x} = \mathbf{s}$.

A compact expression of FIM is revealed by the following [139]

Theorem 20 *In the ICA model* (2.1) *and under Assumptions 1)-3), the FIM $\mathcal{I}_\theta$ of $\theta = \mathrm{vec}(\mathbf{W}^T)$ is a $n^2 \times n^2$ block matrix with (i,j)-block being equal to $n \times n$ matrix:*

$$\mathcal{I}_\theta[i,j] = \begin{cases} \lambda_i \mathbf{a}_i \mathbf{a}_i^T + \kappa_i \sum_{\substack{l=1 \\ l \neq i}}^{n} \mathbf{a}_l \mathbf{a}_l^T & \text{if } i = j \\ \mathbf{a}_j \mathbf{a}_i^T & \text{if } i \neq j \end{cases}$$

The whole $n^2 \times n^2$ matrix $\mathcal{I}_\theta$ can be constructed using the above $n \times n$ blocks $\mathcal{I}_\theta[i,j]$.

Using Theorem 20, a simple and compact expression for the inverse of the FIM can now be presented [139].

Theorem 21 *In the ICA model* (2.1) *and under Assumptions 1)-3) and denoting $\theta = \mathrm{vec}(\mathbf{W}^T)$, $\mathcal{I}_\theta^{-1}$ exists and is a $n^2 \times n^2$ block matrix with (i,j)-block being equal to a $n \times n$ matrix:*

$$\mathcal{I}_\theta^{-1}[i,j] = \begin{cases} \frac{1}{\lambda_i} \mathbf{w}_i \mathbf{w}_i^T + \sum_{\substack{l=1 \\ l \neq i}}^{n} \frac{\kappa_l}{\kappa_i \kappa_l - 1} \mathbf{w}_l \mathbf{w}_l^T & \text{if } i = j \\ -\frac{1}{\kappa_i \kappa_j - 1} \mathbf{w}_j \mathbf{w}_i^T & \text{if } i \neq j \end{cases}$$

Note that diagonal blocks $\mathcal{I}_\theta^{-1}[i,i]$ give the CRB for an unbiased estimator $\hat{\mathbf{w}}_i$ of the de-mixing vector $\mathbf{w}_i$:

$$\mathrm{cov}\{\hat{\mathbf{w}}_i\} \geqslant \frac{1}{n} I_\theta^{-1}[i,i] \tag{8.24}$$

for $i = 1, \ldots, n$. Theorem 21 shows that the CRB depends on the distributions of s_i only through the scalars κ_i and λ_i for $i = 1, \ldots, n$. Theorem 21 also implies that only one of s_i's can be Gaussian: if the first and second component, say, are Gaussian, then $\kappa_1 = \kappa_2 = 1$ and $\kappa_2/(\kappa_1\kappa_2 - 1)$ is not defined. Still, even in this case, any other block $\mathcal{I}_\theta^{-1}[i,i]$ for $i \geq 3$ exists (since the denominators $\kappa_i\kappa_l - 1$, $i \neq l \in \{1, \ldots, n\}$, do not vanish), indicating that all the remaining rows of $\mathbf{W}$ expect the first two can be consistently estimated. That is, the presence of two Gaussian sources does not eliminate the possibility to recover the other sources.

In ICA, the performance of the separation is often investigated via (see also Section 9.1 and in particular eq. (9.2))

$$\hat{\mathbf{Q}} = (\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_n)^T = \hat{\mathbf{W}}\mathbf{A} \tag{8.25}$$

since the estimated i-th source is $\hat{\mathbf{s}}_i = \hat{\mathbf{w}}_i^T \mathbf{x} = \hat{\mathbf{q}}_i^T \mathbf{s} = \sum_{j=1}^{n} \hat{q}_{ij} s_j$. Thus, $\hat{q}_{ij}$ and $\mathrm{var}\{\hat{q}_{ij}\} = E\left\{\hat{q}_{ij}^2\right\}$ for $i \neq j$ represent the magnitude and the

average power of interference of th source in the estimated i-th source signal. Since $E\{\hat{q}_{ij}\} = 1$, the variance $\mathrm{var}\{\hat{q}_{ij}\}$ reflects how accurately the presence of i-th source itself is estimated. The CRB for $\hat{\vartheta} = \mathrm{vec}(\hat{\mathbf{Q}}^T)$ is independent of the parameter $\mathbf{W}$ as it is a nonsingular linear transformation of $\hat{\theta} = \mathrm{vec}(\hat{\mathbf{W}}^T)$, i.e., $\hat{\vartheta} = (\mathbf{I} \otimes \mathbf{A}^T)\hat{\theta}$, where $\otimes$ denotes the *Kronecker product*: for any matrix $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \otimes \mathbf{A}$ is a block matrix with (i,j)-block being equal to $a_{ij}\mathbf{B}$. Therefore, $\mathrm{cov}(\hat{\vartheta}) = (\mathbf{I} \otimes \mathbf{A}^T)\mathrm{cov}(\hat{\theta})(\mathbf{I} \otimes \mathbf{A})$, which by (8.17) and (8.23) indicate that

$$\mathrm{cov}\left(\hat{\vartheta}\right) \geq n^{-1}\left(\mathbf{I} \otimes \mathbf{A}^T\right)\mathcal{I}_\theta^{-1}\left(\mathbf{I} \otimes \mathbf{A}\right) = n^{-1}\mathcal{I}_I^{-1} \tag{8.26}$$

where $\mathcal{I}_I$ denotes the value of $\mathcal{I}_\theta$ at $\theta = \mathrm{vec}(\mathbf{I})$ (i.e., at $\mathbf{W} = \mathbf{I}$). Hence, $\mathrm{cov}(\hat{\mathbf{q}}_i) \geq n^{-1}\mathcal{I}_I^{-1}[i,i]$, and Theorem 21 gives the following bounds:

$$\begin{array}{l} \delta_i = \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} \mathrm{var}\left(\hat{q}_{ij}\right) \geq n^{-1} \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} \frac{\kappa_j}{\kappa_i\kappa_j - 1} \\ \mathrm{var}\left(\hat{q}_{ii}\right) \geq n^{-1}\frac{1}{\lambda_i} \end{array} \tag{8.27}$$

where δ_i may be interpreted as the average power of interfering source signals to the estimated i-th source.

The fact that the CRB for elements of $\hat{\mathbf{Q}}$ is independent of $\mathbf{A}$ is in agreement with the *equivariance property* [35] shared by many ICA estimators (see Section 7.7.1).

To be more specific, let $\hat{\mathbf{W}} = \hat{\mathbf{W}}(\mathbf{X}_n)$ be an estimator of $\mathbf{W}$ based upon i.i.d. data set $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ from the ICA model (2.1). Thus, the $n \times m$ data matrix $\mathbf{X}_n$ can be factored as $\mathbf{X}_n = \mathbf{A}\mathbf{S}_n$, where $\mathbf{S}_n = (\mathbf{s}_1, \ldots, \mathbf{s}_n)$ is an i.i.d. data set distributed as $\mathbf{s}$. Equivariant estimator satisfies $\hat{\mathbf{W}}(\mathbf{X}_n) = \hat{\mathbf{W}}(\mathbf{S}_n)\mathbf{A}^{-1}$ and thus $\hat{\mathbf{Q}} = \hat{\mathbf{W}}(\mathbf{X}_n)\mathbf{A} = \hat{\mathbf{W}}(\mathbf{S}_n)$ is independent of $\mathbf{A}$. This property is nicely reflected in the above derived bound (8.26) for $\hat{\mathbf{Q}}$ [60].

The problem of the procedure described in eq. (8.27) is that the compact formulation of the Cramèr-Rao Lower Bound is calculated since the numerical quantities κ_i and λ_i which are defined on the score functions of the original source s_i. Unfortunately we do not know the original sources s_i and so the score functions φ_i. But if we adopt the flexible solution, i.e. spline functions, as described in Section 7, we have the result that the activation functions h_i are an estimate of the cdfs of the estimated sources. In this way the *flexibility* of spline functions allows the proposed network to make an actual estimate of the CRLB.

Form the i-th activation function, it is possible to derive an estimate $\tilde{\varphi}_i(u_i)$ of the true score functions $\varphi_i(s_i)$:

$$\tilde{\varphi}_i\left(u_i\right) = \frac{h_i''\left(u_i\right)}{h_i'\left(u_i\right)} \tag{8.28}$$

In this way we can calculate an estimated version of κ_i and λ_i in eqs. (8.21) and (8.22) respectively. From Lemma 5 we derive

$$\varphi_{\mathbf{s}}(\mathbf{s}) = (\mathbf{WA})^T \varphi_{\mathbf{u}}(\mathbf{u}) \tag{8.29}$$

From eqs. (8.29) and (2.3) and considering the case in which we have no permutation ambiguity ($\mathbf{P} = \mathbf{I}$ in (2.3)), it is possible to write the following relation for the single marginal score function

$$\tilde{\varphi}_i(s_i) = \alpha_i \tilde{\varphi}_i(u_i) \tag{8.30}$$

where α_i is the non-zero entry in the matrix $\mathbf{WA}$. Moreover for the i-th source of the vector $\mathbf{s} = (\mathbf{WA})^{-1}\mathbf{u}$ holds that:

$$s_i = [(\mathbf{WA})^{-1}]_i \, u_i \tag{8.31}$$

where $[(\mathbf{WA})^{-1}]_i$ is the i-th row of the $\mathbf{WA}$ product matrix. If we considering again that only the scaling ambiguity is present, then eq. (8.31) can be rewritten as

$$s_i = \alpha_i^{-1} u_i \tag{8.32}$$

The relations in eqs. (8.30) and (8.32) allow us to express an estimate $\tilde{\kappa}_i$ and $\tilde{\lambda}_i$ of the two previous quantities κ_i and λ_i:

$$\tilde{\kappa}_i = E\left\{\tilde{\varphi}_i^2\left(u_i\right)\right\} \tag{8.33}$$

$$\tilde{\lambda}_i = E\left\{\tilde{\varphi}_i^2\left(u_i\right) u_i^2\right\} - 1 \tag{8.34}$$

Therefore substituting the quantities in eqs. (8.33) and (8.34) in formulas (8.27) we are able to perform the Cramér-Rao Lower Bound for the elements in $\hat{\mathbf{Q}}$.

If we perform the MMI approach (see Section 7.4.2) we have not the activation function $h_i(u_i)$ but a direct estimate $\tilde{\varphi}_i(u_i)$ of the score functions by the *Least Mean Square (LMS)* algorithm, through eqs. (7.35) and (7.36), and so we can reuse the (8.33) and (8.34) in formulas (8.27) for calculating the CRLB.

9 Results

—The most difficult part of a trip is to cross the doorway
P. Terentius Varro

THIS section collects some experimental results in order to demonstrate the effectiveness of our complex domain approach both in linear and nonlinear environment. Examples of application of the new Riemannian metrics are also proposed.

9.1 Performance evaluation

There are no standardized method to realize performance analysis but there exist several indexes and algorithms less or more diffused in literature; the best choice is to select the index adequate to the problem and to the mixing/de-mixing environment.

In order to compare performances we adopt the index introduced in [168], which evaluates the presence of the desired signal for each channel. In this way the quality of separation of the k-th separated output can be defined by the *Signal to Interference Ratio (SIR)* as (see Figure 9.1)

$$SIR\,(k) = 10\log\left[E\left\{\left(|u|_{\sigma(k),k}\right)^2\right\} \Big/ E\left\{\sum_{i\neq k}\left(|u|_{\sigma(k),i}\right)^2\right\}\right]. \qquad (9.1)$$

In eq. (9.1) $u_{i,k}$ is the i-th output signal when only the k-th input signal s_k is present, while $\sigma(k)$ is the output channel corresponding to the k-th input. This index is able to provide the evaluation of separation results without considering the particular mixing/de-mixing structure but only the original sources and the recovered signals. This is a very attractive characteristic

which leads this index to be used to compare the performance of separation also in case of different mixing/de-mixing models.

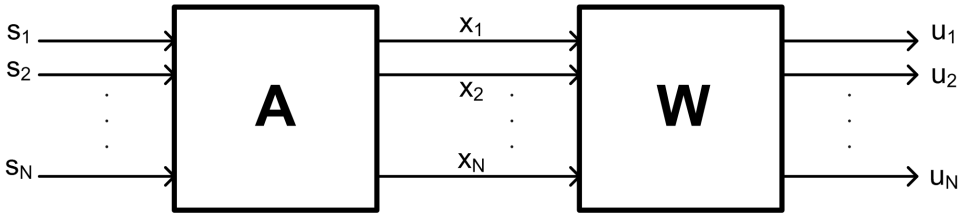

Fig. 9.1: Evaluation of SIR index

Another way to evaluate the performance of an algorithm in linear case only is to analyze the matrix product **WA** which has to be close to the product of a diagonal matrix and a permutation matrix. Thus according to the desired solution to BSS problem, only one element in each row and column can be substantially a non zero element. Let us assume q_{ij} the generic element of the matrix $\mathbf{Q} = \mathbf{WA}$, we can define the following performance index [5]

$$S = \sum_{i=1}^{N} \left(\frac{\sum_{k=1}^{N} |q_{ik}|^2}{\max_p \left[|q_{ip}|^2 \right]} - 1 \right) + \sum_{k=1}^{N} \left(\frac{\sum_{i=1}^{N} |q_{ik}|^2}{\max_p \left[|q_{pk}|^2 \right]} - 1 \right). \tag{9.2}$$

The index in eq. (9.2) is a non-negative number and it is equal to zero only for perfect separation. For other details see eq. (2.6) and Lemma 2.

9.2 Performance test in linear mixing environment

The subsection is dedicated to the evaluation of algorithm proposed to solve the BSS problem in linear environment. The free parameters of algorithm are the following ones: the number $N_{\mathbf{h}}$ of spline control points used in eq. (6.23), the learning rate $\eta_{\mathbf{W}}$ involved in the adaptation of the entries in the de-mixing matrix, the learning rate $\eta_{\mathbf{h}}$ involved in the adaptation of the spline control points and the number of runs (or epochs) of the algorithm n_R.

9.2.1 First Test

In the first experiment we adopt the algorithm with the bi-dimensional AF in eq. (6.32), using the following four complex sources: s_1 is a 8-PSK (Phase Shift Keying) modulation, s_2 is a uniform random noise and s_3 is a 4-QAM modulation [154].

The mixing environment in eq. (4.1) is:

$$\mathbf{A} = \begin{bmatrix} 1.5 - j0.5 & -0.4 + j0.6 & 0.5 + j0.1 \\ 0.5 + j1 & 0.4 + j0.2 & -0.1 + j0.3 \\ 0.2 - j0.4 & -0.6 + j1 & 1.3 + j0.6 \end{bmatrix}$$

while the free parameters are summarized in the following Table 9.1. In

$N_{\mathbf{h}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	n_R
21	$2,3 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	100

Table 9.1: Free parameters in test 1

Figure 9.2 we present the joint pdf of the original signals (the first row), of the mixtures (the second row) and of the separated signals (the third row). In Figure 9.3 we report the performance of the algorithm. Comparing the two performance graphics we can note that the algorithm converge in about 20 epochs and the SIR index for the separated signals is between 25 dB and 40 dB; the performance are confirmed by the Performance Index (9.2) in the second row of figures. We can compare this result with respect the same test done with a fully-complex $tanh(z)$ AF as described in [2] (see Figure 9.4); in this case the training was stopped after 1000 epochs. We do not report the scatter plot of signals because these are very similar to the first ones. We can note that in the case of the fully-complex $tanh\,(z)$ AF the convergence is slower (about 200 epochs), while the performance is between 15 dB and 25 dB.

While the quality of separation is very similar, the flexible approach is faster.

9.2.2 Second Test

In the second experiment we adopt the algorithm with the bi-dimensional AF in eq. (6.32), using the following four complex sources: s_1 is a 8-PSK (Phase Shift Keying) modulation, s_2 is a 16-QAM (Quadrature Amplitude Modulation) modulation, s_3 is a 4-QAM modulation and s_4 is a uniform random noise [154].

The mixing environment in eq. (4.1) is:

$$\mathbf{A} = \begin{bmatrix} 1.5 - j0.5 & -0.4 + j0.6 & 0.5 + j0.1 & 0.1 \\ 0.5 + j & 0.4 + j0.2 & -0.1 + j0.3 & 0.2 + j0.5 \\ 0.2 - j0.4 & -0.6 + j & 1.3 + j0.6 & 0.4 + j \\ 0.2 + j & -0.7 + j0.1 & -0.1 - j0.1 & 0.8 - j0.7 \end{bmatrix}$$

while the free parameters are summarized in the following Table 9.2.

In Figure 9.5 we present the joint pdf of the original signals (the first row), of the mixtures (the second row) and of the separated signals (the third

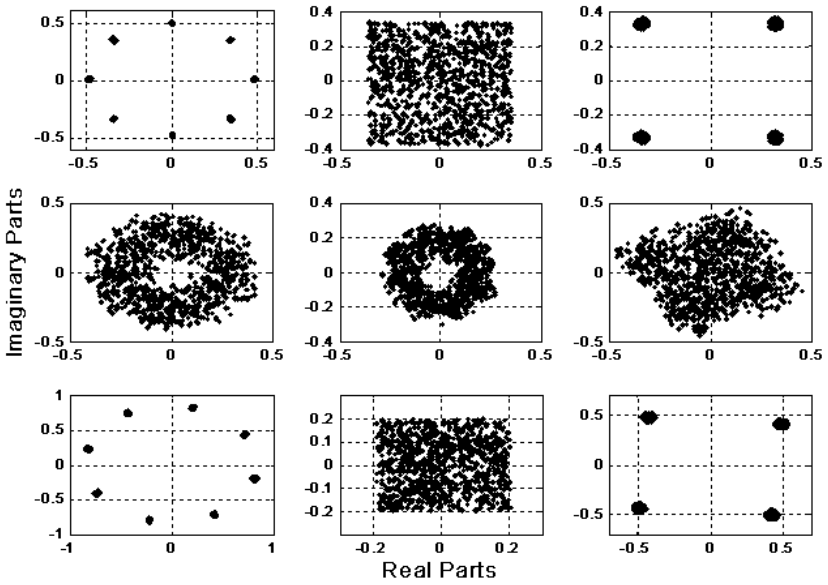

Fig. 9.2: Scatter plot of original signals (first row), mixtures (second row) and separated signals (third row)

$N_{\mathbf{h}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	n_R
21	$2,3 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	100

Table 9.2: Free parameters in test 2

row). In Figure 9.6 we report the performance of the algorithm. Comparing the two performance graphics we can note that the algorithm converge in about 50 epochs and the SIR index for the separated signals is between 26 dB and 37 dB. We can compare this result with respect the same test done with a fully-complex $tanh(z)$ AF as described in [2] (see Figure 9.7); in this case the training was stopped after 1000 epochs. We do not report the scatter plot of signals because these are very similar to the first ones. We can note that in the case of the fully-complex $tanh\,(z)$ AF the convergence is slower (about 200 epochs), while the performance is between 25 dB and 39 dB.

While the quality of separation is very similar, the flexible approach is faster.

9.2.3 Third Test

In the third experiment we adopt the algorithm with the bi-dimensional AF in eq. (6.32) using the following three complex sources: s_1 is a 8-PSK modulation, s_2 is a 4-QAM modulation and s_3 is a uniform random noise.

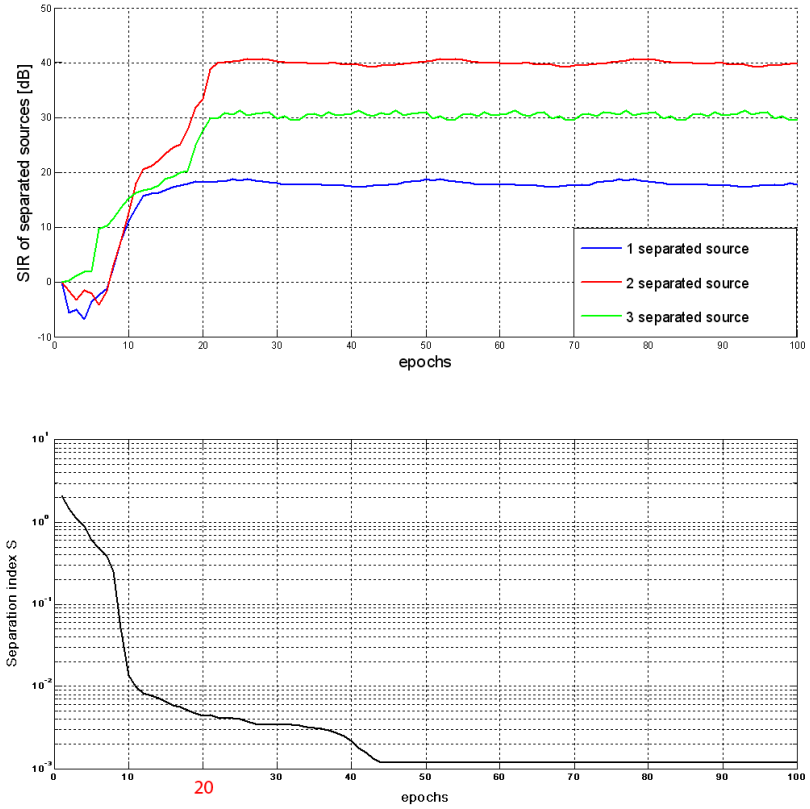

Fig. 9.3: Performance indexes for the first experiment

Then we modified these signals in order to correlate the real and imaginary part, in order to simulate the effect of a communication channel [163]. The correlation between the real and imaginary part is obtained by varying the length M of a moving average *FIR filter (FIRMA)*. In this way each sample of the imaginary part of the k-th signal s_{Ik} is obtained as the mean over M past samples of its real part s_{Rk}:

$$s_{Ik}(n) = \frac{1}{M} \sum_{p=0}^{M-1} s_{Rk}(n-p). \tag{9.3}$$

The mixing environment in eq. (4.1) is:

$$\mathbf{A} = \begin{bmatrix} 1.5 - j0.5 & -0.4 + j0.6 & 0.5 + j0.1 \\ 0.5 + j & 0.4 + j0.2 & -0.1 + j0.3 \\ 0.2 - j0.4 & -0.6 + j & 1.3 + j0.6 \end{bmatrix}$$

while the free parameters are summarized in the following Table 9.3. The correlation coefficients between the real and imaginary part of the original source are shown in Table 9.4

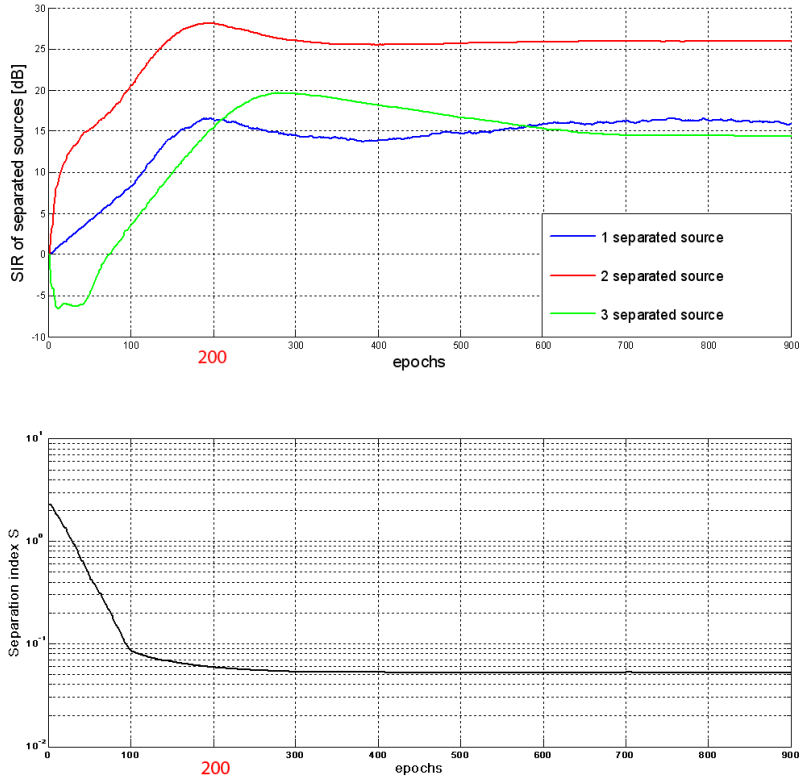

Fig. 9.4: Performance indexes in the case of $tanh(z)$ activation function

$N_{\mathbf{h}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	n_R
21	$2,3 \cdot 10^{-3}$	$5 \cdot 10^{-4}$	100

Table 9.3: Free parameters in test 3

In Figure 9.8 we present the joint pdf of the original signals (the first row), of the mixtures (the second row) and of the separated signals (the third row).

In Figure 9.9 we report the performance index in eq. (9.2) of the algorithm versus the M parameter in eq. (9.3) comparing this result with respect the same test done with the algorithm described in [2] using the $tanh(z)$ AF; in this case the training was stopped after 1000 epochs.

Comparing the two performance graphics we can note that the convergence is more accurate and stable even the parameter M is varied.

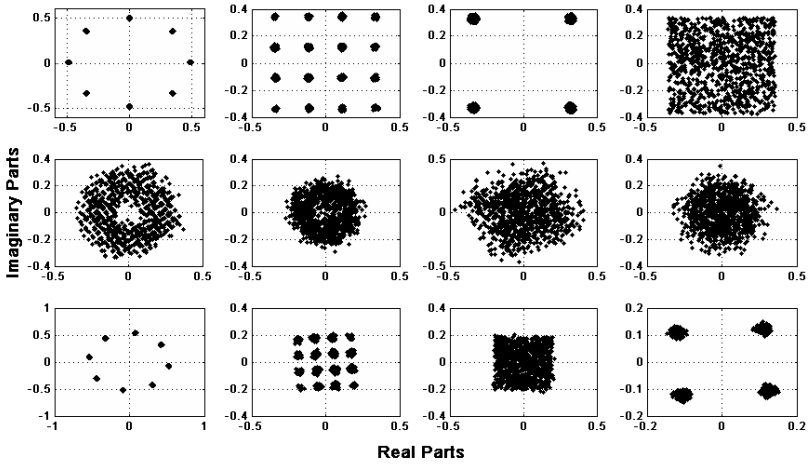

Fig. 9.5: Scatter plot of original signals (first row), mixtures (second row) and separated signals (third row)

M	4-QAM	8-PSK	WN
0	0.0179	0.0155	0.0085
1	0.6950	0.4842	0.7183
5	0.3676	0.3811	0.4092
15	0.2123	0.2713	0.2101
25	0.1317	0.1557	0.1309

Table 9.4: Correlation coefficients vs. M parameter

9.2.4 Fourth Test

A fourth test is performed adopting the three signals involved in the previous example but varying the number $N_{\mathbf{h}}$ of spline control points in the following set of values: $\{13, 16, 21, 31, 61\}$. The other parameters are the same that in Table 9.3.

Figure 9.10 shows the separation index in eq. (9.2) for the five different values of $N_{\mathbf{h}}$. The index profile shows that $N_{\mathbf{h}} = 21$ points is the best choice. If the number of control points is large the quality of separation is bad while if it is too small the index has an oscillatory behavior.

9.2.5 Fifth Test

As fifth test we have proposed a comparison of the Performance Index in eq. (9.2) varying the spline basis: Catmull-Rom spline, B-Spline, Bezier spline and Hermite spline (see eqs. (6.25)-(6.28)).

Figure 9.11 shows that Bezier spline basis and hermite spline basis are not suitable for blind source separation applications, while good results were

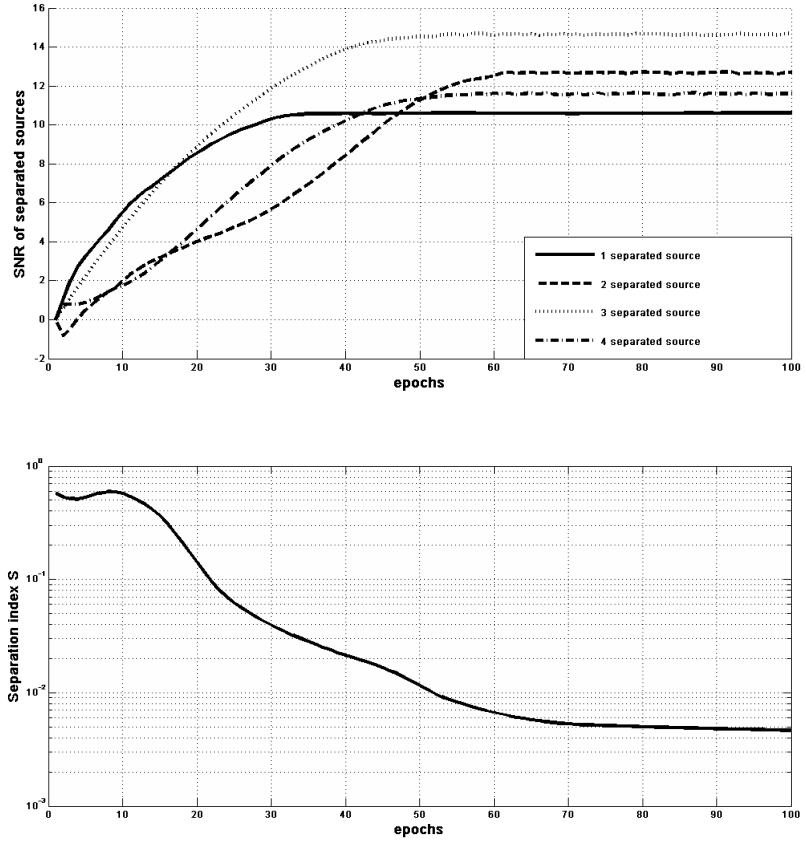

Fig. 9.6: Performance indexes for the second experiment

found in the case of usage of Catmull-Roma or B-Spline basis, even if this latter is sometimes unstable and take the algorithm away the convergence for a certain number of epochs.

9.2.6 Comparison with other approaches

In this experiment we compare the proposed complex and flexible INFOMAX with two different approaches well known in literature: the maximization of the kurtosis [113, 114] and the maximization of Negentropy [137]. These approaches are two counterpart of the *complex maximization of non-Gaussianity (CMN)* algorithm (see section 7.6).

Figure 9.12 shows the results of these two different approaches using the Performance Index in eq. (9.2). The profile of the figure assures that the proposed method is faster and more accurate than the other ones.

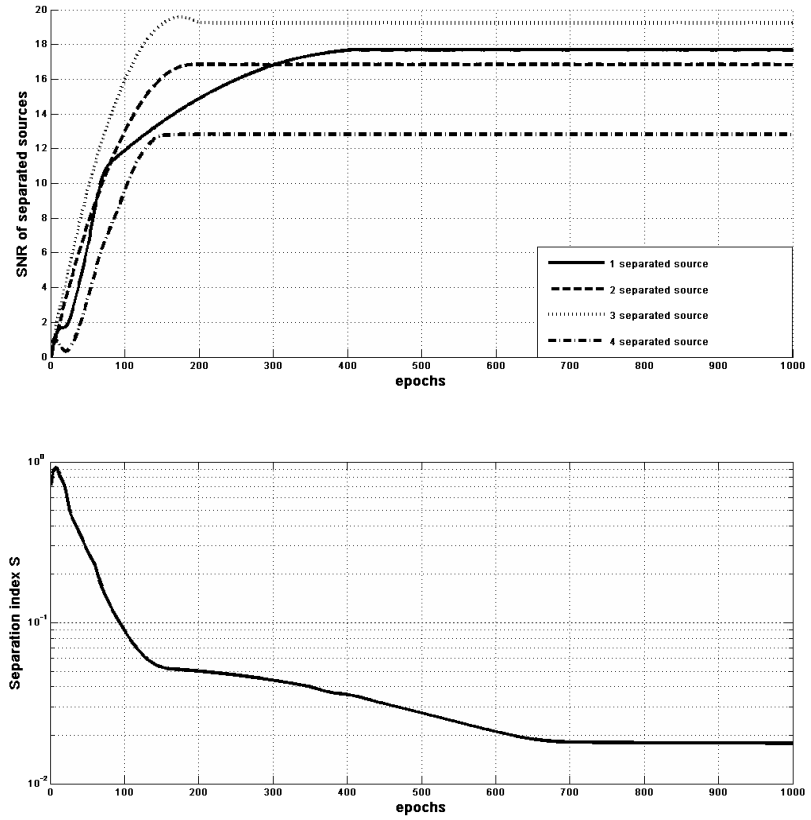

Fig. 9.7: Performance indexes in the case of $tanh(z)$ activation function

9.2.7 Convolved speech signals

In this experimental test we use the following three signals: s_1 is a reader, s_2 is a radio recording and s_3 is a uniform noise. All signals were subsampled at a frequency of 4 kHz. The convolutive environment was transformed in a multidimensional complex-valued and instantaneous source separation problem, as described is Section 3.2.1.

Figure 9.13 shows the SIR Index in eq. (9.1) for the three separated sources. The profile shows that two of the sources are well separated, while the radio recording is still quite confuse.

9.2.8 fMRI signals

In this experimental test we use the 8 fMRI signals, each composed by 61×61 pixels. This environment was transformed in a multidimensional complex-valued and instantaneous source separation problem, as described is Section 3.2.2.

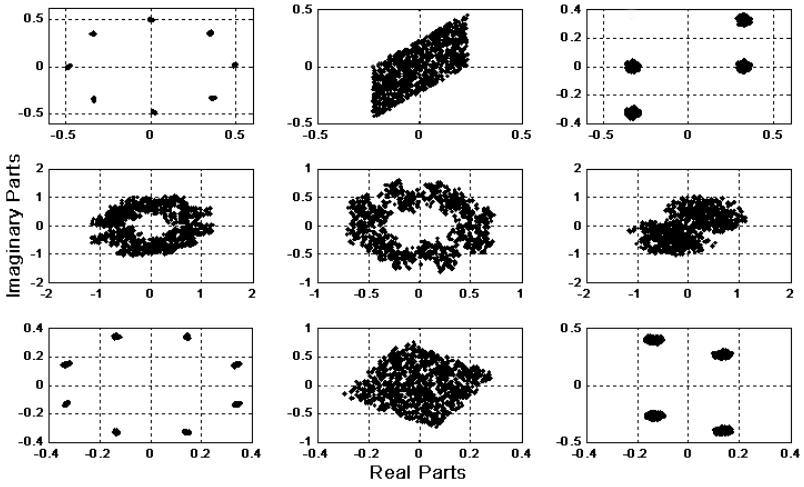

Fig. 9.8: Scatter plot of original signals (first row), mixtures (second row) and separated signals (third row)

Figure 9.14 shows the Performance Index in eq. (9.2) for the separated sources. The profile shows that also in this case the proposed algorithm can reach excellent degree of separation.

9.3 Performance test in nonlinear mixing environment

This subsection is dedicated to the evaluation of the algorithm proposed to solve the BSS problem in nonlinear environment. The free parameters of algorithm are the following ones: the number $N_{\mathbf{h}}$ and $N_{\mathbf{G}}$ of spline control points used for the complex activation functions and the nonlinear compensating functions, the learning rate $\eta_{\mathbf{W}}$ involved in the adaptation of the entries in the de-mixing matrix, the learning rate $\eta_{\mathbf{h}}$ involved in the adaptation of the activation functions, the learning rate $\eta_{\mathbf{G}}$ involved in the adaptation of the nonlinear compensating functions and the number of runs (or epochs) of the algorithm n_R.

9.3.1 First Test

For the first test we adopt the algorithm with the mono-dimensional AF in eq. (6.31) using a 4-QAM signal, a uniform random signal and a PSK signal. The mixing environment in eq. (4.3) is:

$$\mathbf{A} = \begin{bmatrix} 0.90 - j0.30 & -0.24 + j0.36 & 0.30 + j0.06 \\ 0.30 + j0.60 & 0.24 + j0.12 & -0.06 + j0.18 \\ 0.12 - j0.24 & -0.36 + j0.60 & 0.78 + j0.36 \end{bmatrix},$$

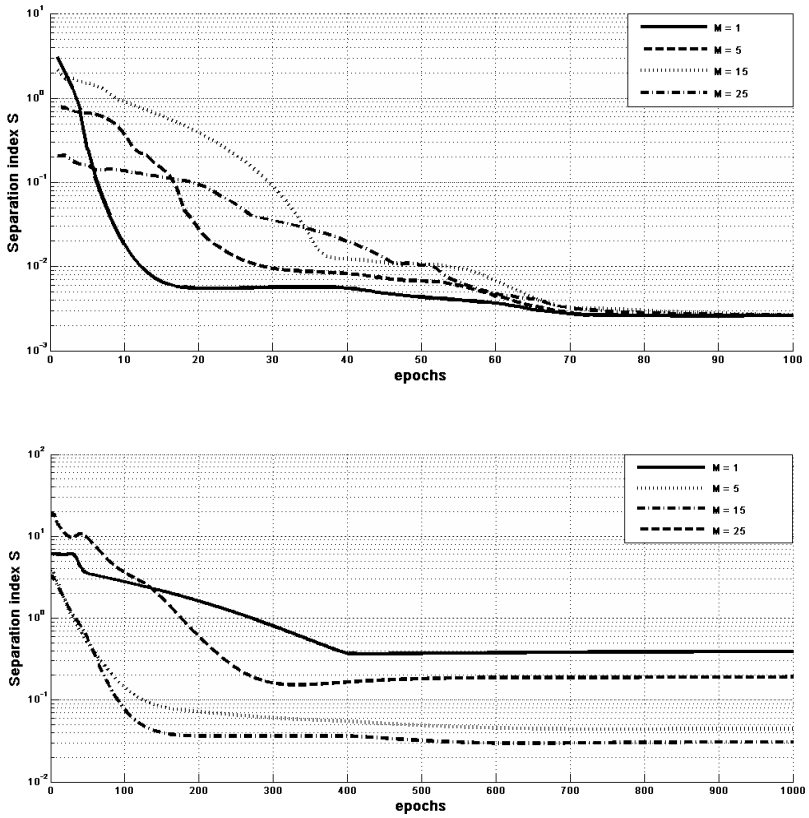

Fig. 9.9: Performance index vs. the length M of the MA filter: the flexible generalized splitting activation function (top) and a $tanh(z)$ activation function (bottom)

$$\mathbf{F}[\mathbf{v}] = \begin{bmatrix} f_1[v_1] = \left(v_{R1} + 0.7v_{R1}^3\right) + j\left(v_{I1} + 0.7v_{I1}^3\right) \\ f_2[v_2] = \left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) + j\left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) \\ f_3[v_3] = \left(v_{R3} + 0.7v_{R3}^3\right) + j\left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) \end{bmatrix}$$

while the free parameters are summarized in the following Table 9.5.

$N_{\mathbf{h}}$	$N_{\mathbf{G}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	$\eta_{\mathbf{G}}$	n_R
31	31	$5 \cdot 10^{-5}$	$5 \cdot 10^{-6}$	$5 \cdot 10^{-6}$	400

Table 9.5: Free parameters in test 1

The effectiveness of the separation is evidenced in Figure 9.15 that shows the joint pdf of the original sources (first row), of the nonlinear mixture (second row) and finally of the separated signals (third row).

Figure 9.16 shows that after about 200 epochs the training became stable and more accurate. So the profiles of the separation index $SIR(k)$ in eq. (9.1) for each channel assures the effectiveness of the learning.

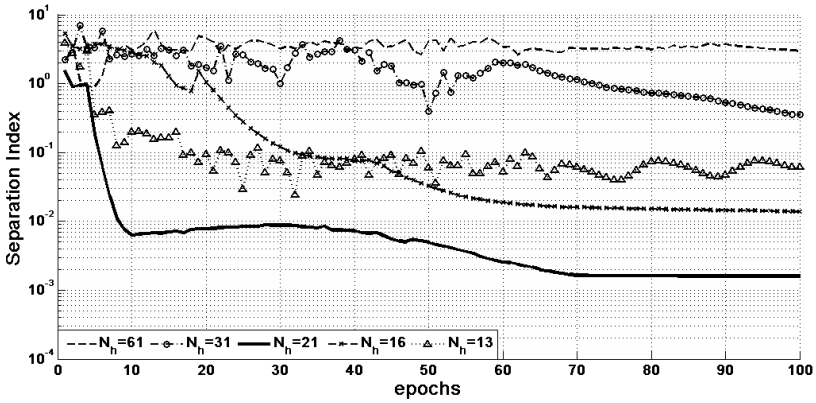

Fig. 9.10: Performance index vs. the number of spline control points

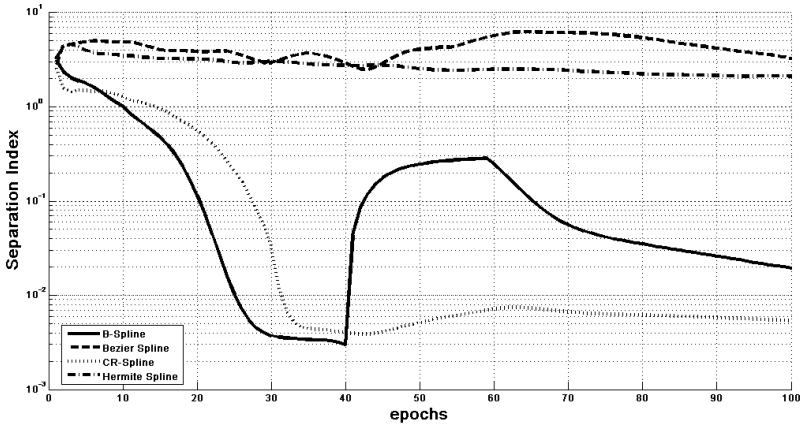

Fig. 9.11: Performance Index for different spline basis

9.3.2 Second Test

The effectiveness of the separation is evidenced in Figure 9.17 that shows the joint pdf of the original sources (first row), of the nonlinear mixture (second row) and finally of the separated signals (third row).

Figure 9.18 shows that after about 100 epochs the training became stable and more accurate. So the profiles of the separation index $SIR(k)$ in eq. (9.1) for each channel assures the effectiveness of the learning.

9.3.3 Third Test

A second test is done with 16-QAM signal, a 8-PSK signal and an artificial Bernoulli's lemniscate signal. The choice of this particular and strange signal is due to the fact that we are interested in test our algorithm in signal with a

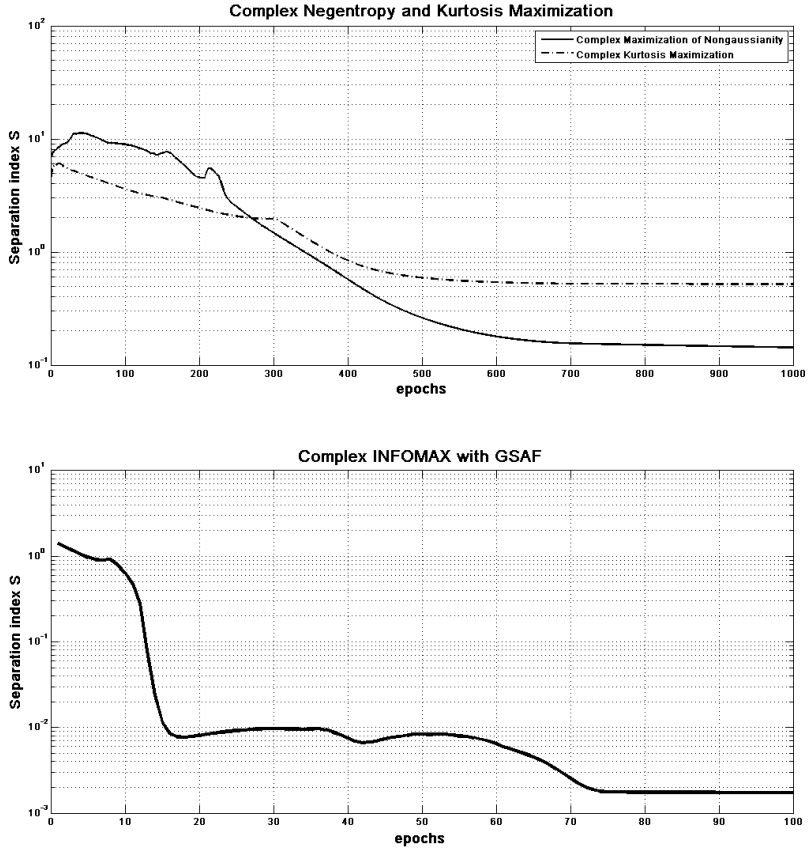

Fig. 9.12: Different ICA approaches: Negentropy and Kurtosis maximization (up) and Flexible INFOMAX (bottom)

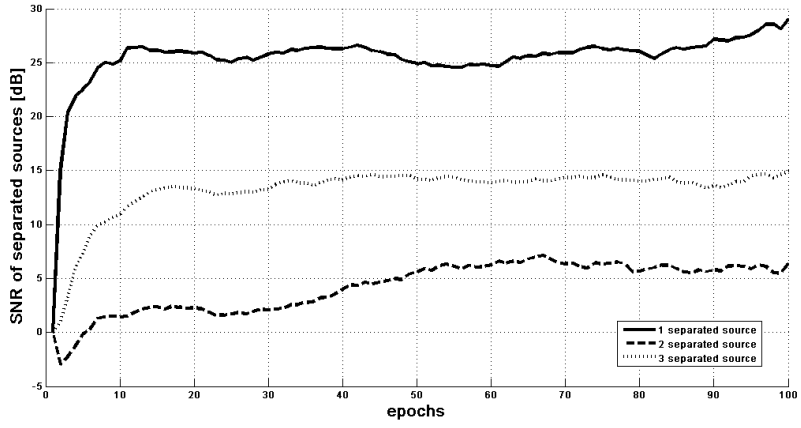

Fig. 9.13: SIR of convolved speech signals

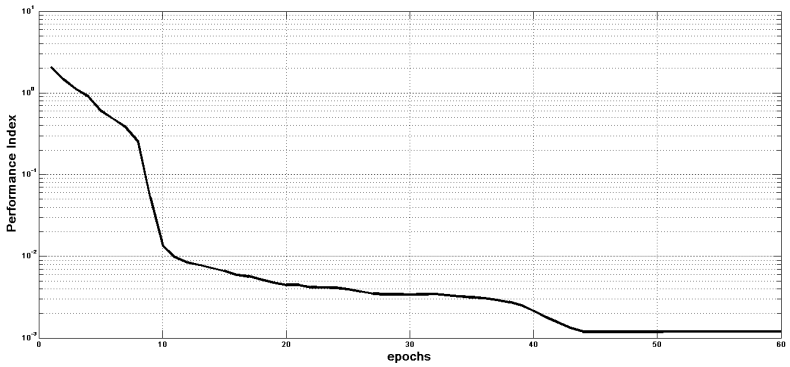

Fig. 9.14: Performance Index of fMRI signals

real and imaginary part strongly correlated.

The mixing environment in eq. (4.3) and the free parameters are the same values of the previous test, see Table 9.5.

The effectiveness of the separation is evidenced in Figure 9.19 that shows the joint pdf of the original sources (first row), of the nonlinear mixture (second row) and finally of the separated signals (third row).

Figure 9.20 shows that after about 200 epochs the training became stable and more accurate. So the profiles of the separation index $SIR(k)$ in eq. (9.1) for each channel assures the effectiveness of the learning.

9.3.4 Recovering of the nonlinearities

The aim of this experimental test is to show as the proposed network is able to recover the estimate $g_i(\bullet)$ of the inverse of the distorting function $f_i(\bullet)$.

Figure 9.21 proposes the graphic of the distorting nonlinear function $f_1(v_1[n])$ of the first test in Section 9.3.1

$$f_1\left[v_1\right] = \left(v_{R1} + 0.7v_{R1}^3\right) + j\left(v_{I1} + 0.7v_{I1}^3\right)$$

in the first row, the graphic of the recovered inverse $g_1(x_1[n])$ function in the second row and the composition of the two previous functions $g \circ f$ in the third row. As we can see, the composition is quite linear and so the network was able to recover the inverse of the nonlinear f function quite well.

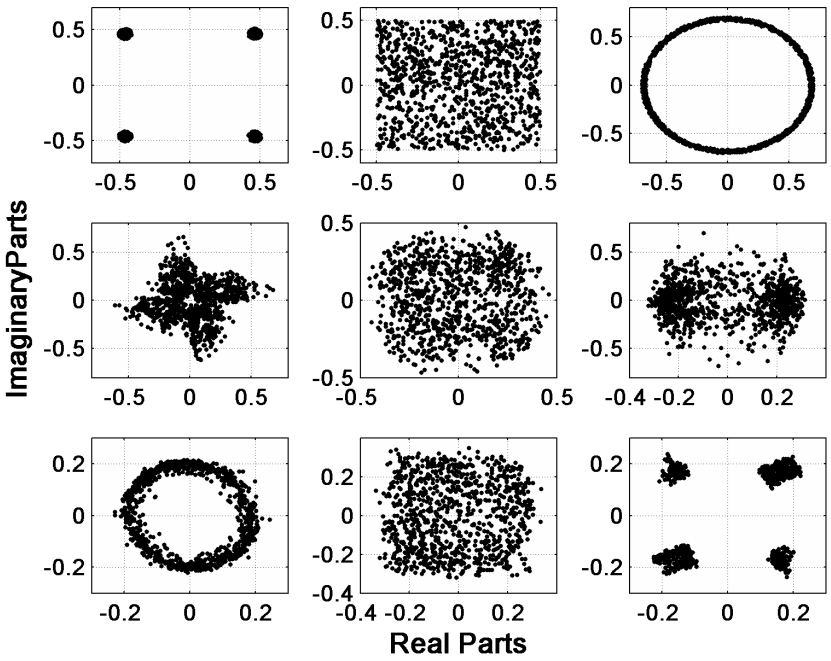

Fig. 9.15: Scatter plot of original sources (first row), mixtures (second row) and separated sources (third row)

9.3.5 Absolutely degenerate mixing matrix

In this test we adopt the following *absolutely degenerate* mixing matrix (see Definition 14)

$$\mathbf{A} = \begin{bmatrix} 1.5 - j0.5 & -0.4 - j0.4 & 1.5 - j0.5 \\ -0.5 + j0.4 & 1.4 + j0.9 & 0.5 - j0.4 \\ 0.2 - j0.4 & -0.6 + j0.8 & 0.2 - j0.4 \end{bmatrix}$$

As stated in Section 2.7, when we have an absolutely degenerate mixing matrix we can at least recover the source only, but not the nonlinearities. This fact is demonstrated by Figure 9.22 where an estimate of the inverse $g_1(x_1[n])$ function is represented. As we can see this figure shows a quite linear function, which is not the true inverse of the original distorting function.

9.3.6 Comparisons with other approaches

Last experimental test collects the results of comparison between the algorithm here introduced for PNL mixtures (named *Flexible Complex Post Non-Liner ICA* or *FC-PNLICA*) and another algorithm. Unfortunately in literature there is not any algorithm working on PNL mixtures in the complex case. In this sense we can compare the results only with an algorithm

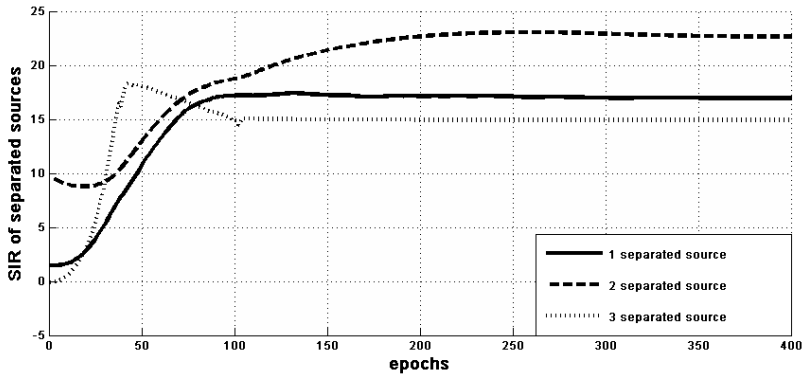

Fig. 9.16: Separation index during training for the first experiment in PNL mixing environment

working on linear mixtures. The algorithm chosen for the comparison is that described in a previous section and proposed by [2] that uses the ETFs (here named *Complex Linear ICA* or *C-LICA*). The comparison has been performed both in linear mixing environment and in the nonlinear one. Here, the Separation Index has been evaluated for both algorithms modifying the shape of the nonlinear distorting functions. Just to simplify the exposition of the results, parametric non linear function have been used for this test:

$$\mathbf{F}\left[\mathbf{v},\alpha,\beta,\gamma\right]=\left[\begin{array}{c} f_1\left[v_1,\alpha\right]=\left(v_{R1}+\alpha v_{R1}^3\right)+j\left(v_{I1}+\alpha v_{I1}^3\right) \\ f_2\left[v_2,\beta\right]=\left(v_{R2}+\beta\tanh\left(3v_{R2}^3\right)\right)+j\left(v_{R2}+\beta\tanh\left(3v_{R2}^3\right)\right) \\ f_3\left[v_3,\gamma\right]=\left(v_{R3}+\gamma v_{R3}^3\right)+j\left(v_{R2}+\gamma\tanh\left(3v_{R2}^3\right)\right) \end{array}\right]. \tag{9.4}$$

The following table collects the Separation Index in eq. (9.1) after 600 epochs with the same learning rate of test 9.3.1 but random starting conditions with different value of $[\alpha,\beta,\gamma]$ in eq. (9.4) such that $[\alpha,\beta,\gamma]=\left\{\ [0,0,0]\quad[0.4,0.4,0.4]\quad[0.7,0.7,0.7]\quad[1,1,1]\ \right\}$.

$SIR(k)$	$\alpha=\beta=\gamma\leq 0$	$\alpha=\beta=\gamma\leq 0.4$	$\alpha=\beta=\gamma\leq 0.7$	$\alpha=\beta=\gamma\leq 1$
FC-PNLICA	[77.70,49.84,101.10]	[20.14,40.15,26.70]	[30.94,14.53,8.94]	[26.65,12.09,6.59]
C-LICA	[80.16,39.84,120.38]	[6.02,24.55,36.91]	[25.02,1.40,-7.72]	[-6.37,-6.58,-9.52]

Table 9.6: Comparison of SIR values for two algorithms

The results collected in Table 9.6 show how the separation performance of the algorithm FC-PNLICA and the C-LICA are comparable if the parameter of non-linear distortion are $\alpha=\beta=\gamma\leq 0.4$. With higher level of distortion the C-LICA is no more able to reach the separation but the FC-PNLICA is able to guarantee still good results.

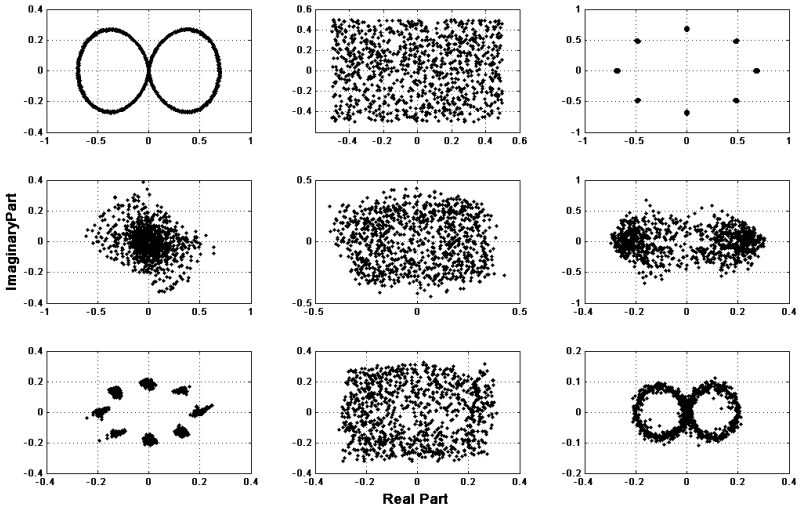

Fig. 9.17: Scatter plot of original sources (first row), mixtures (second row) and separated sources (third row)

9.4 Performance test with new riemannian learning rules

In order to test our architecture, we used a 8-PSK (Phase Shift Keying) signal, a uniform noise signal and a 4-QAM Quadrature Amplitude Modulation) signal, each with 1000 samples. In all the experiments we use the following parameters: the learning rate for the $\mathbf{W}$ matrix is set to $\eta_W = 0.0001$, while the learning rate for the spline control points is set to $\eta_Q = 0.0005$. The spline step is $\Delta = 0.3$ and the algorithm runs for 100 epochs. The results are mean over 30 runs.

The results of the first experimental test are summarized in the following table 9.7 [167].

We perform a second test using a 16-PSK (Phase Shift Keying) signal, a 4-QAM (Quadrature Amplitude Modulation) signal and a 16-QAM signal, each with 1000 samples. In all the experiments we use the previous parameters. The results of this test are summarized in the following table 9.8.

Table 9.7 shows that the standard stochastic gradient provides a quite good separation performance, but the learning is slow: it is stable after about 50 epochs. Better is the solution with the standard natural gradient (right natural gradient): the value of SIR is higher and the learning is faster (about 25 epochs). More performance is the use of right/left and right/right natural gradient: good SIR and very fast (about 15 and 23 epochs respectively); while worst is the case of left and left/left natural gradient: bad SIR, but quite fast (better than the standard stochastic gradient) as provided in section 7.7.1.

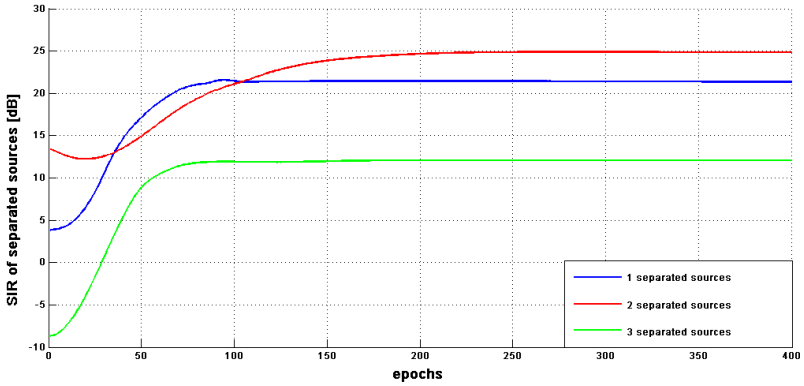

Fig. 9.18: Separation index during training for the second experiment in PNL mixing environment

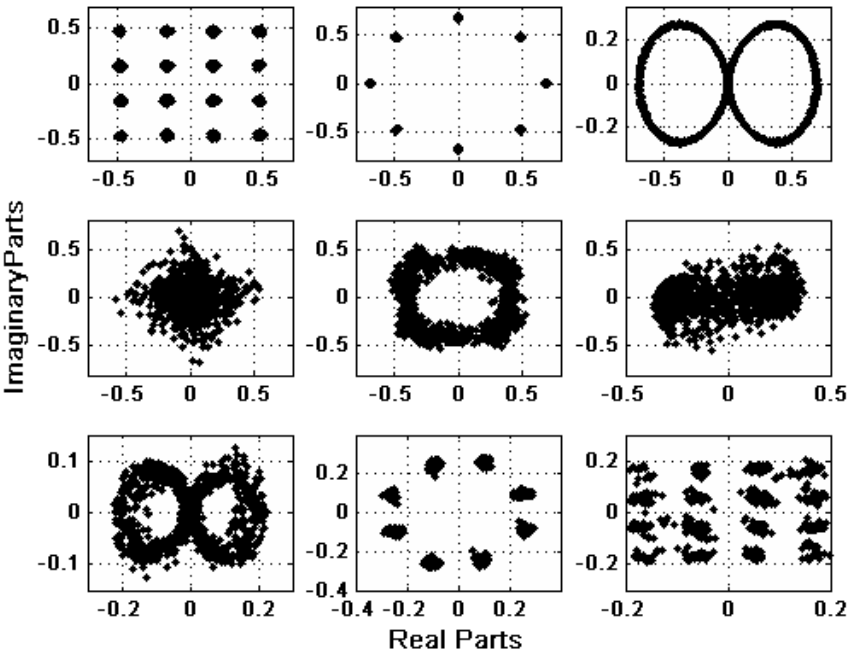

Fig. 9.19: Scatter plot of original sources (first row), mixtures (second row) and separated sources (third row)

Softly worst are the results in the second test (table 9.8), due to the presence of two similar distributed signals (8-PSK and 16-PSK), but the trend is similar to the first test.

The use of the generalized splitting activation function realized with the bi-dimensional spline functions improve more the convergence speed with

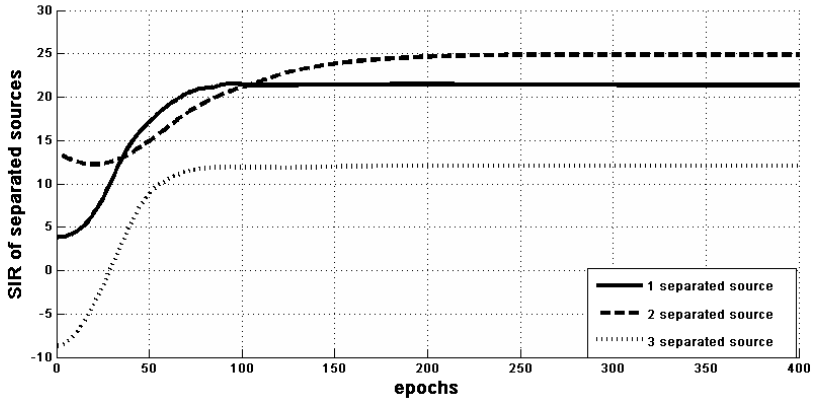

Fig. 9.20: Separation index during training for the third experiment in PNL mixing environment

Gradient	**SIR** (dB)			**Sep. Index**	**Convergence**
	s_1	s_2	s_3	$\times 10^{-3}$	(epochs)
Standard	37,99	8,11	22,96	238	50
Right	50,48	40,05	34,35	2,29	25
Left	42,73	18,55	24,02	33	30
Right/left	39,24	33,14	37,26	1,63	15
Left/left	9,37	9,56	9,55	752	20
Right/right	45,41	34,45	40,40	1,041	23

Table 9.7: Results of the test 1

respect a fix function, comparing our tests with the ones in [180].

Gradient	**SIR** (dB)			**Sep. Index**	**Convergence**
	s_1	s_2	s_3	$\times 10^{-3}$	(epochs)
Standard	30,02	17,08	20,40	62	60
Rigth	35,95	42,61	29,85	3,46	35
Left	34,71	10,87	5,98	45	25
Right/left	33,22	13,70	10,11	2,03	35
Left/left	17,68	21,57	12,06	380	15
Right/right	23,87	38,10	23,21	2,10	30

Table 9.8: Results of the test 2

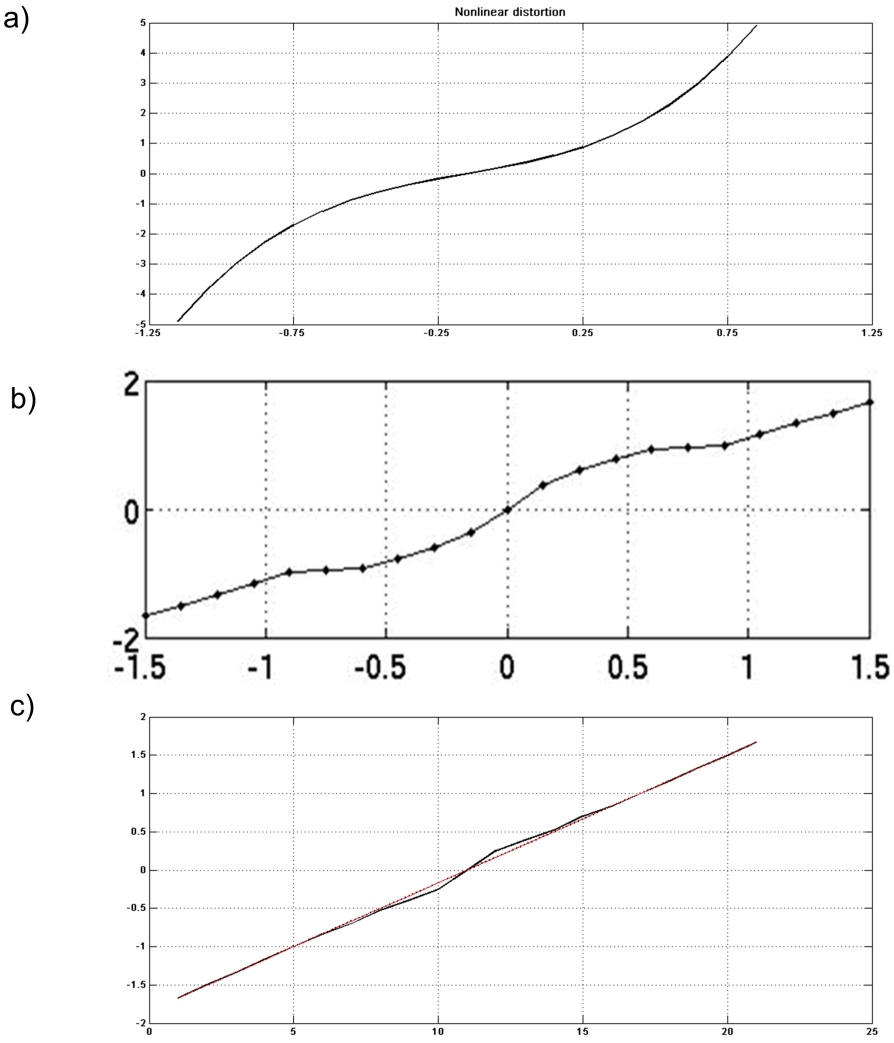

Fig. 9.21: The distorting nonlinearity a); its inverse estimation b) and the composition of the two functions c)

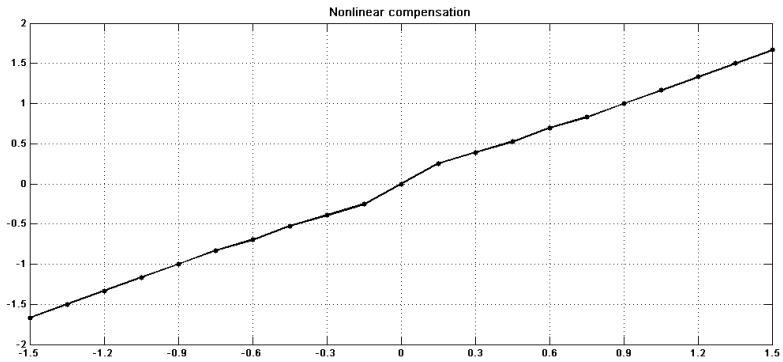

Fig. 9.22: A nonlinearity in the case of an absolutely degenerate mixing matrix

10

Conclusions

—Looks like we've made it,
Look how far we've come, my baby.
You are still the one that I love,
The only one I dream of.
S. Twain

CONSIDERING the evolution of ICA algorithms in solving BSS problems it is important to underline that, although several studies exist in the real domain, in the case of the complex domain the state of art it is not so advanced. This thesis collects a first trial to enhance the state of art of mixing environment for which ICA algorithms can provide a solution.

In this thesis a novel complex model of mixing environment has been introduced and described even in linear that in nonlinear mixing model. The BSS problem in this new environment is solved by exploiting an ICA-based algorithm. The proposed approach extends the well-known *INFOMAX* algorithm, based on the *Maximum Entropy* and *Minimal Mutual Information* approaches, to the complex domain and is based on the use of flexible spline networks to perform local on-line estimation of the activation functions and nonlinear compensating functions.

The usefulness of a complex representation is also investigated and three interesting examples are shown: speech mixtures in *reverberant* environment, *functional Magnetic Resonance Imaging (fMRI)* signals and *band-pass* telecommunication signals. The disagreement of such a representation is also emphasized in paragraph 3.1.1.

It is deeply analyzed the dichotomy between analyticity and boundedness in the choice of the nonlinear *activation function* (AF) and the advantages of adopting a *splitting solution* or, at best, a *generalized splitting solution*. In this case the complex activation function is obtained as the complex sum of two

real-valued functions, mono-dimensional or bi-dimensional respectively.

Moreover the INFOMAX algorithm is *equivalent* to the ICA, if and only if, the shape of the activation function is identical to the shape of the cumulative density function (cdf) of the corresponding original source. This fact justifies the adopting of a *flexible solution*: the activation function, or better the real and imaginary part - splitting solution - of the activation function, are realized by *spline function*. Splines are a superposition of a certain number of pieces of polynomial functions. In this way the shape of the complex activation functions is iteratively changed during the learning process.

A complete derivation of the learning algorithms, both in linear and nonlinear domain, is then presented. It is shown that these learning rules are formally very simple and of simple implementation, although their analytical derivations are very hard. Different cost functions are utilized: *joint entropy*, *mutual information* (or Kullback-Leibler divergence) and *Renyi's entropy*. In a first phase the classical *stochastic gradient* algorithm is implemented in the optimization process. In order to overcome to the problem of this kind of gradient rule, a *natural gradient* learning rule is also proposed. The natural gradient assures a faster and more accurate convergence of the algorithm. In addition four *new recent* Riemannian algorithms were proposed. These algorithms provide an even more faster behavior. The existence and uniqueness of the solution for the proposed algorithm in the nonlinear environment is demonstrated also.

In the last chapter of this thesis three interesting topics are introduced. First of all, it is shown that the spline function assures an *universal behavior* in the Blind Source Separation problem. The *flexibility* of this kind of function makes splines suitable for separation of both super-gaussian and sub-gaussian sources. A second problem in BSS problem is the scaling ambiguity that in complex domain gives birth to a *phase ambiguity* too. It is shown that a splitting solution is able to recovery the phase information. The calculation of the *Cramèr-Rao Lower Bound (CRLB)* is then investigated.

Quality of the separation has been evaluated in terms of *separation index* (SIR) and *performance index* in a number of experimental tests, both in linear and in nonlinear environment. The profiles of these graphics demonstrate the effectiveness of the proposed approach and the proposed algorithms.

10.1 Future research directions

This is an initial work on efficient and flexible neural architectures for Blind Signal Separation in complex environment. Necessary extensions of the research in BSS to complex domain environments via ICA approach must be addressed to improvements of the mixing models. A first necessary extension is the employment of the generalized splitting function in the nonlinear case, even if the learning rules could seem to be very hard. Secondly

it could be interesting to extend the flexible methods to other approaches, i.e. the complex maximization of non-Gaussianity (CMN) algorithm.

Another fundamental extension is an approach based on a complex and convolutive environment in order to better model real world application, as telecommunication applications.

More advances in new Riemannian metrics and in the formalization of a compact formulation for the Cramèr-Rao Lower Bound in the nonlinear environment are needed, in order of better understanding the nonlinear domain.

Recently I am working on a Post Nonlinear (PNL) extension to the problem of *Blind Source Extraction (BSE)*, which is not reported in this thesis. In BSE problem only one source is recovered from mixtures at every turn of the algorithm, according to its statistical property. The BSE is very helpful in such applications where a great number of mixtures are available but one is interested in a very small number of sources, like biomedical applications (ECG, EEG, etc.). It is very interesting and useful to place these new results in the general framework of the complex BSS.

Bibliography

[1] T. Adali. Why a nonlinear solution for a linear problem? In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 157–165, 23-25 Aug. 1999.

[2] T. Adali, T. Kim, and V. D. Calhoun. Independent component analysis, by complex nonlinearities. In *Proc. ICASSP 2004*, volume 5, pages 525–528, Montreal, Canada, 2004.

[3] T. Adali and H. Li. A practical formulation for computation of complex gradients and its application to maximum likelihood ica. In *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume II, pages 633–636, Honolulu, Hawaii, April 2007.

[4] J. H. Ahlberg, E. N. Nilson, and J. L. Walsh. *The theory of splines and their applications*. Academic Press, 1967.

[5] S. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996.

[6] S-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[7] S.-I. Amari, T.-P. Chen, and A. Cichocki. Stability analysis of adaptive blind source separation. *Neural Networks*, 10(8):1345–1351, 1997.

[8] B. Ans, J. Hérault, and C. Jutten. Adaptive neural architectures: detection of primitives. In *Proceedings of COGNITIVA'85*, pages 593–597, Paris, France, 1985.

[9] H. Aoki, M. R. Azimi-Sadjadi, and Y. Kosugi. Image association using a complex-valued associative memory model. *Transactions of the IEICE*, 83A:1824–1832, 2000.

[10] A. Arcangeli, S. Squartini, and F. Piazza. An alternative natural gradient approach for ica based learning algorithms in blind source separation. In *Proceedings of EUSIPCO 2004*, pages 593–596, Vienna, Austria, 2004.

[11] M. Babaie-Zadeh. *On blind source separation in convolutive and nonlinear mixtures*. PhD thesis, Universitt'e technologique SHARIF, 2002.

[12] M. Babaie-Zadeh, C. Jutten, and K. Nayebi. A geometric approach for separating pnl mixtures. In *Proceedind of ICA 2004*, 2004.

[13] R. E. Barnhill and R. F. Riesenfeld, editors. *Computer Aided Geometric Design*. New York Academic, New York, 1974.

[14] A. J. Bell and T. J. Sejnowski. *Advances in Neural Information Processing Systems*, volume 7, chapter A non-linear information maximization algorithm that performs blind separation, pages 467–474. The MIT Press, Cambridge, MA, 1995.

[15] A. J. Bell and T. J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[16] A. Belouchrani and M. G. Amin. Blind source separation based on time-frequency signal representations. *IEEE Transaction on Signal Processing*, 46(11):2888–2897, November 1998.

[17] A. Belouchrani and W. Ren. Blind carrier phase tracking with guaranteed global convergence. *IEEE Transaction on Signal Processing*, 45(7):1889–1894, 1997.

[18] A. Belouchrani and Wei Ren. New formulation of the carrier phase tracking problem. In *Signals, Systems and Computers, 1996. 1996 Conference Record of the Thirtieth Asilomar Conference on*, pages 1186–1190vol.2, 3-6 Nov. 1996.

[19] N. Benvenuto, M. Marchesi, F. Piazza, and A. Uncini. Nonlinear satellite radio links equalized using blind neural network. In *Proc. of ICASSP*, volume 3, pages 1521–1524, May 1991.

[20] N. Benvenuto and F. Piazza. On the complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 40(4):967–969, April 1992.

[21] E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex valued signals. *International Journal of Neural Systems*, 10(1), 2000.

[22] D. H. Brandwood. A complex gradient operator and its application in adaptive array theory. *IEE Proceedings, F: Communications, Radar and Signal Processing*, 130(1):11–16, 1983.

[23] G. Bunkheila, A. Cirillo, M. Scarpiniti, R. Parisi, and A. Uncini. Interfacce acustiche intelligenti in ambienti confinati. In *Proc. of AES Italian Section, Annual Meeting*, Parma, Italy, November 23-25 2007.

[24] G. Burel. Blind separation of sources: a nonlinear neural algorithm. *Neural Networks*, 5(6):937–947, 1992.

[25] V. Calhoun, T. Adali, and G. Pearlson. Independent component analysis applied to fmri data: a generative model for validating results. In *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, pages 509–518, 10-12 Sept. 2001.

[26] V. D. Calhoun and T. Adali. Complex infomax: Convergence and approximation of infomax with complex nonlinearities. In *Proc. NNSP*, pages 307–316, Switzerland, 2002.

[27] V. D. Calhoun, T. Adali, L. K. L. K. Hansen, J. Larsen, and J. J. Pekar. Ica of functional mri data: an overview. In *4th INternational Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 281–288, Nara, Japan, April 2003.

[28] V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar. A method for making group inferences from functional mri data using independent component analysis. *Hum. Brain Map.*, 14:140–151, 2001.

[29] V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar. On complex infomax applied to complex fmri data. In *Proc. ICASSP*, Orlando, FL, 2002.

[30] V. D. Calhoun, T. Adali, G. D. Pearlson, P. C. Van Zijl, and J. J. Pekar. Independent component analysis of fmri data in the complex domain. *Magn. Reson. Med.*, 48:180–192, 2002.

[31] X-R. Cao and R-W. Liu. General approach to blind source separation. *TRANSACTIONS ON SIGNAL PROCESSING, IEEE*, 44(3):562–571, March 1996.

[32] J.-F. Cardoso. Blind identification of independent signals. In *Proc.Workshop on Higher-Order Specral Analysis*, Vail, Colorado, 1989.

[33] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025, Oct. 1998.

[34] J.-F. Cardoso and P. Comon. Independent component analysis, a survey of some algebraic methods. In *Circuits and Systems, 1996. ISCAS '96., 'Connecting the World'., 1996 IEEE International Symposium on*, volume 2, pages 93–96vol.2, 12-15 May 1996.

[35] J.-F. Cardoso and B. H. Laheld. Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 45(2):3014–3030, 1996.

[36] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[37] J.F. Cardoso. Source separation using higher order moments. In *Proc. ICASSP*, pages 2109–2112, 1989.

[38] J.F. Cardoso and T. Adali. The maximum likelihood approach to complex ica. In *Proc. of Int. Conf. on Acoustic, Speech and Signal Processing*, volume V, pages 673–676, Toulouse, France, May 2006.

[39] E. Catmull and R. Rom. *Computer-Aided Geometric Design*, chapter A class of local interpolating splines, pages 317–326. New York Accademic, 1974.

[40] N. Charkani and Y. Deville. Optimization of the asymptotic performance of time domain convolutive sources separation algorithm. In *Proceedings of the ESANN*, pages 273–278, Bruges, Belgium, 1997.

[41] T. Chen, H. Chen, and R. Liu. Approximation capability in C($\mathbb{R}^n$) by multilayer feedforward networks and related problems. *IEEE Trans. Neural Networks*, 6:25–30, Jan 1995.

[42] S. Choi, A. Cichocki, and S. Amari. Flexible independent component analysis. *Journal of VLSI Signal Processing - Systems for, Image and video Technology*, X(1), May 2000.

[43] L. O. Chua, C. A. Desoer, and E. S. Kuh. *Linear and Nonlinear Circuits*. McGraw-Hill, 1987.

[44] A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing*. John Wiley, 2002.

[45] A. Cichocki and L. Moszczynski. New learning algorithm for blind separation of sources. *Electronics Letters*, 28(21):1986–1987, 8 Oct. 1992.

[46] A. Cichocki and R. Unbehauen. Robust neural networks with online learning for blind identification and blind separation of sources.

IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 43(11):894–906, Nov. 1996.

[47] A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 30(17):1386–1387, 18 Aug. 1994.

[48] T. L. Clarke. Generalization of neural networks to the complex plane. In *Proc.. of IJCNN*, volume II, pages 435–440, 1990.

[49] P. Comon. Separation of stochastic processes. In *Proc. Workshop on Higher-Order Specral Analysis*, pages 174–179, Vail, Colorado, 1989.

[50] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[51] P. Comon and O. Grellier. Non-linear inversion of underdetermined mixtures. In *Proc. ICA'99*, pages 461–465, Aussois, France, Jan 1999.

[52] P. Comon, C. Jutten, and J. Herault. Blind separation of sources, part ii: Problems statement. *Signal Processing*, 24:11–20, 1991.

[53] N. Correa, T. Adali, Yi-Ou Li, and V.D. Calhoun. Comparison of blind source separation algorithms for fmri using a new matlab toolbox: Gift. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 5, pages v/401–v/404Vol.5, 18-23 March 2005.

[54] C. Cottin, I. Gavrea, H. H. Gonska, D. P. Kacsò, and D-X. Zhou. Global smoothness preservation and the variation-diminishing property. *Journal of Inequality & Applications*, 4:91–114, 1999.

[55] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John WIley & Sons, Inc., 2nd edition edition, 2006.

[56] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[57] G. Darmois. Analyse générale des liaisons stochastiques. *Rev. Inst. Internationale Statist.*, 21:2–8, 1953.

[58] Carl De Boor. *A practical guide to spline*. Applied Mathematical Science. Springer, 2001.

[59] D. L. Donoho. *Applied Time Series Analysis II*, chapter On minimum entropy deconvolution, pages 565–608. Academic Press, 1981.

[60] E. Doron, A. Yeredor, and P. Tichavský. A cramér-rao-induced bound for blind separation of stationary parametric gaussian sources. *IEEE Signal Processing Letter*, 14(6):417–420, 2007.

[61] J. Eriksson and V. Koivunen. Blind identifiability of class of nonlinear instantaneous ica models. In *XI European Signal Processing Conference EUSIPCO2002*, pages 7–10, Toulouse, France, 3-6 September 2002.

[62] J. Eriksson and V. Koivunen. Identifiability and separability of linear ica models revisited. In *4-th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 23–27, Nara, Japan, April 2003.

[63] J. Eriksson and V. Koivunen. Complex-valued ica using second order statistics. In *IEEE Workshop on Machine Learning for Signal Procwssing*, pages 183–191, 2004.

[64] J. Eriksson and V. Koivunen. Identifiability, separability and uniqueness of linear ica models. *Signal Processing Letters, IEEE*, 11(7):601–604, July 2004.

[65] J. Eriksson and V. Koivunen. Blind separation of a class of nonlinear ica models. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2005*, pages 5890–5893, 23–26 May 2005.

[66] J. Eriksson and V. Koivunen. Complex random vectors and ica models: identifiability, uniqueness, and separability. *Information Theory, IEEE Transactions on*, 52(3):1017–1029, March 2006.

[67] S. Fiori. Blind signal processing by the adaptive activation function neuron. *Neural Networks*, 13(6):597–611, August 2000.

[68] S. Fiori, A. Uncini, and F. Piazza. Neural blind separation of complex sources by extended apex algorithm (eapex). In *Proc. of IEEE ISCAS'99*, volume V, pages 627–630, Orlando (USA), June 1999.

[69] G. M. Georgiou and C. Koutsougeras. Complex domain backpropagation. *IEEE Trans. On Circuit and System II*, 39(5):330–334, May 1992.

[70] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[71] Su Lee Goh, Z. Babic, D. Popovic, T. Tanaka, and D. Mandic. Complex-valued neural network schemes for online processing of wind signal. In *Neural Network Applications in Electrical Engineering, 2004. NEUREL 2004. 2004 7th Seminar on*, pages 249–253, 23-25 Sept. 2004.

[72] Su Lee Goh, D.H. Popovic, and D.P. Mandic. Complex-valued estimation of wind profile and wind power. In *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, volume 3, pages 1037–1040Vol.3, 12-15 May 2004.

[73] G. H. Golub and C. F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press, 3rd edition, 1996.

[74] S. Guarnieri, F. Piazza, and A. Uncini. Multilayer feedforward networks with adaptive spline activation function. *IEEE Trans. on Neural Network*, 10(3):672–683, May 1999.

[75] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, 3rd edition, 1996.

[76] S. Haykin. *Neural Networks, a comprehensive foundation*. Prentice-Hall, 2nd edition, 1999.

[77] S. Haykin, editor. *Unsupervised Adaptive Filtering, Volume1: Blind Source Separation*. John Wiley & Sons, Inc, 2000.

[78] L. He, T. Thaiupathump, and S. A. Kassam. Blind separation of complex i/q independent sources with phase recovery. *Signal Processing Letter, IEEE*, 12(5):419–422, May 2005.

[79] II Hild, K.E., D. Erdogmus, and J. Principe. Blind source separation using renyi's mutual information. *Signal Processing Letters, IEEE*, 8(6):174–176, June 2001.

[80] A: Hiroe. Solution of permutation problem in frequency domain ica, using multivariate probability density functions. In *Proceeding of ICA*, LNCS, pages 601–608, 2006.

[81] A. Hirose. Continuous complex-valued back-propagation learning. *Electronics Letters*, 28:1854–1855, 1992.

[82] A. Hirose. *Complex valued neural networks, Theories and Applications*. World Scientific, 2003.

[83] J. C. Holladay. Smoothest curve approximation. *Math. Tables Aids Computation*, 11:233–243, 1957.

[84] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[85] K. Hornik, M. Stinchombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[86] A. Hyvärinen. A family of fixed-point algorithms for independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 3917–3920, Munich, Germany, 1997.

[87] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 10(3):626–634, May 1999.

[88] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John WIley & Sons, Inc., 2001.

[89] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[90] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[91] A. Hyvärinen and P. Pajunen. On existence and uniqueness of solutions in nonlinear independent component analysis. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks (IJCNN'98)*, volume 2, pages 1350–1355, 1998.

[92] G. Johnson. Questioning 'distinction without a difference' debate. *IEEE Signal Processing Magazine*, 12(6):19, November 1995.

[93] G. Johnson. The complex (and circular) argument continues. *IEEE Signal Processing Magazine*, 13(5):42–44, September 1996.

[94] C. Jutten and J. Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[95] C. Jutten and J. Karhunen. Advances in nonlinear blind sources separation. In *4th International Symposium on Independent Component Analisys and Blind Signal Separation (ICA2003)*, Nara, Japan, April 2003.

[96] C. Jutten and A. Taleb. Source separation: from dusk till dawn. In *Proceeding of 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA2000)*, pages 15–26, Helsinki, Finland, 2000.

[97] A. Kagan, Y. Linnik, and C. Rao. *Characterization Problems in Mathematical Statistics*. Probability and Mathematical Statistics. Wiley, New York, 1973.

[98] J. Karhunen, S. Malaroiu, and M. Ilmoniemi. Local linear independent component analysis based on clustering. *Int. J. of Neural Systems*, 10(6), 2000.

[99] T. Kim and T. Adali. Fully complex backpropagation for constant envelope signal processing. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 1, pages 231–240vol.1, 11-13 Dec. 2000.

[100] T. Kim and T. Adali. Approximation by fully complex mlp using elementary transcendental activation functions. In *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, pages 203–212, 10-12 Sept. 2001.

[101] T. Kim and T. Adali. Complex backpropagation neural network using elementary trascendental activation functions. In *Proc. of IEEE ICASSP*, volume II, pages 1281–1284, 2001.

[102] T. Kim and T. Adali. Fully complex multilayer perceptron for nonlinear signal processing. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 32:29–43, Aug.-Sep. 2002.

[103] T. Kim and T. Adali. Universal approximation of fully complex feedforeward neural network. In *Proc. Of IEEE ICASSP'02*, volume 1, pages 973–976, Orlando, FL, 2002.

[104] T. Kim and T. Adali. Approximation by fully-complex multilayer prceptrons. *Neural Computation*, 15(7):1641–1666, July 2003.

[105] G. J. Klir. *Uncertainty and Information (Foundations of Generalized Information Theory)*. John WIley & Sons, Inc., 2006.

[106] J.-L. Lacoume and P. Ruiz. Sources identification: a solution based on cumulants. In *Proc. IEEE ASSP Workshop*, Minneapolis, Minnesota, 1988.

[107] R.H. Lambert. *Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures*. PhD thesis, University of Southern California, May 1996.

[108] T. Lee. Nonlinear approaches to independent component analysis. In *Proceedings of the American Institute of Physics*, 1999.

[109] T. Lee, M. Girolami, and T. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11:417–441, 1999.

[110] T. Lee, B. Koehler, and R. Orglmeister. Blind separation of nonlinear mixing models. In *Proceeding of IEEE NNSP*, page 406Ű415, 1997.

[111] T. W. Lee, M. Girolami, A. J. Bell, and T. J. Sejnowsky. A unifying information-theoretic framework for independent component analysis. *In. International Journal of Computer and Mathematics with application computation*, 39(11):11–21, 1999.

[112] H. Leung and S. Haykin. The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, 39(9):2101–2104, Sept. 1991.

[113] H. Li and T. Adali. Gradient and fixed-point complex ica algorithms based on kurtosis maximization. In *Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pages 85–90, Sept. 2006.

[114] H. Li and T. Adali. Complex-valued adaptive signal processing using nonlinear functions. *EURASIP Journal on Advances in Signal Processing*, 2008:1–9, 2008.

[115] Y. Lomnitz and A. Yeredor. A blind ml-scheme for blind source separation. In *IEEE Workshop on Statistical Signal Processing (SSP)*, St. Louis, MO, Sept. 28-Oct. 1 2003.

[116] E. Lukacs. *Characteristic Functions*. Griffin, London, U.K., 2nd edition, 1970.

[117] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus*. Wiley, Chichester, U.K., 1999.

[118] S. Makino, T-W. Lee, and H. Sawada, editors. *Blind Speech Separation*. Springer, 2007.

[119] D. P. Mandic and S. L. Goh. *Complex Valued Nonlinear Adaptive Filters: A Neural Network Approach*. Wiley, 2008.

[120] D. P. Mandic, M. Golz, A. Kuh, D. Obradovic, and T. Tanaka, editors. *Signal processing techniques for Knowledge Extraction and Information Fusion*. Springer, 2007.

[121] D.P. Mandic, S. Javidi, G. Souretis, and V.S.L. Goh. Why a complex valued solution for a real domain problem. In *Proc. IEEE Workshop on Machine Learning for Signal Processing*, pages 384–389, 27–29 Aug. 2007.

[122] A. I. Markusevic. *Elementi di Teoria delle Funzioni Analitiche*. Editori Riuniti, 1988. in italian.

[123] G. C. Marques and L. B. Almeida. Separation of nonlinear mixtures using pattern repulsion. In J. F. Cardoso, C. Jutten, and T. Loubaton, editors, *Proceedings of ICA '99*, pages 277–283, 1999.

[124] M. Marsden and I. J. Schoenberg. On variation diminishing spline approximation methods. Technical report, Defense Technical Information Center OAI-PMH Repository (United States), 1998.

[125] H. Mathis and S.C. Douglas. On the existance of universal nonlinearities for blind source separation. *IEEE Transaction on Signal Processing*, 50(5):1007–1016, May 2002.

[126] K. Matsuaka, M. Ohya, and M. Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

[127] M. J. McKeown and T. J. Sejnowski. Independent component analysis of fmri data: Examining the assumptions. *Hum. Brain Map.*, 6:368–372, 1998.

[128] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, Pa, USA, 2000.

[129] L. Molgedey and H. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72:3634–3636, 1994.

[130] R. Mukai, H. Sawada, S. Araki, and A. Makino. Frequency-domain blind source separation of many speech signals using near-field and far-field models. *EURASIP Journal on Applied Signal Processing*, 2006:1–13, 2006.

[131] R. Mukai, H. Sawada, S. Araki, and S. Makino. Blind source separation of 3-d located many speech signals. In *2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 9–12, New Paltz, NY, October 16-19 2005.

[132] J. A. Mumford and T. Nichols. Modeling and inference of multisubject fmri data. *IEEE Eng. Med. Biol. Mag.*, 25(2):42–51, Mar./Apr 2006.

[133] J.-P. Nadal and N. Parga. Non-linear neurons in the low noise limit: a factorial code maximises information transfer. *Network*, 5:565–581, 1994.

[134] F. Neeser and J. Massey. Proper complex random processes with applications to information theory. *IEEE Transaction on Infarmation Theory*, 39(4):1293–1302, Jul. 1993.

[135] C. Nikias and M. Shao. *Signal Processing With α-Stable Distributions and Applications*. Wiley, New York, 1995.

[136] M. Novey and T. Adali. Adaptable nonlinearity for complex maximization of nongaussianity and a fixed-point algorithm. In *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pages 79–84, Sept. 2006.

[137] M. Novey and T. Adali. Complex ica by negentropy maximization. *IEEE Transaction on Neural Networks*, 19(4):596–609, April 2008.

[138] S. Ogawa, S.D.W. Tank, R. Menon, J.M. Ellermann, S.G. Kim, H. Merkle, and K. Ugurbil. Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging. *Proc. Nat. Acad. Sci.*, 89(13):5951–5955, 1992.

[139] E. Ollila, H.J. Kim, and V. Koivunen. Compact cramér-rao bound expression for independent component analysis. *IEEE Transaction on Signal Processing*, 56(4):1421–1428, April 2008.

[140] E. Ollila, V. Koivunen, and J. Eriksson. On the cramér-rao bound for the constrained and unconstrained complex parameters. In *Proceedings of 5th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM2008)*, pages 414–418, 2008.

[141] A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

[142] P. Pajunen, A. Hyvarinen, and J. Karhunen. Nonlinear blind source separation by self-organizing maps. In *Progress in Neural Information Processing, Proc. of the International Conference on Neural Information Processing (ICONIP'96)*, pages 1207–1210, Hong Kong, 1996.

[143] P. Pajunen and J. Karhunen. Blind separation of binary sources with less sensors than sources. In *Proc. ICNN'97*, pages 1994–1997, Houston, TX, June 1997.

[144] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1991.

[145] D.-T. Pham and P. Garat. Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Transaction On Signal Processing*, 45(7):1712–1725, Jul. 1997.

[146] D. T. Pham, D. Garrat, and C. Jutten. Separation of mixture of independent sources through maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774, Brussel, Belgium, August 1992.

[147] F. Piazza, A. Uncini, and M. Zenobi. Neural networks with digital lut activation function. In *Proceedings of IJCNN*, volume II, pages 1401–1404, Nagoya, Japan, 1993.

[148] B. Picinbono. On circularity. *IEEE Transactions on Signal Processing*, 42(12):3473–3482, Dec. 1994.

[149] B. Picinbono. Second-order complex random vectors and normal distributions. *IEEE Transactions on Signal Processing*, 44(10):2637–2640, Oct. 1996.

[150] B. Picinbono. Too complex to be real. *IEEE Signal Processing Magazine*, 13(4):18–19, July 1996.

[151] A. Pierani, F. Piazza, M. Solazzi, and A. Uncini. Low complexity adaptive non-linear function for blind signal separation. In *Proc. of IEEE IJCNN2000*, volume 3, pages 333–338, Como (Italy), 2000.

[152] J. C. Principe and D. Xu. Information-theoretic learning using renyi's quadratic entropy. In *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 407–412, Aussois, France, 1999.

[153] J. C. Principe, D. Xu, and J. W. Fisher III. *Unsupervised Adaptiv Filtering, Volume1: Blind Source Separation*, chapter Information-Theoretic Learning, pages 265–319. Wiley, 2000.

[154] J. G. Proakis. *Digital Communications*. McGraw-Hill, 3rd edition, 1995.

[155] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice-Hall, 1996.

[156] S. Pupolin and L. J. Greenstein. Performance analysis of digital radio links with nonlinear transmit amplifiers. *IEEE Joum. on Selected Areas in Communications*, SAC-5(3):534–456, 1987.

[157] R. Remmert. *Theory of complex functions*. Springer, 1991.

[158] A. Renyi. *Selected Papers of Alfred Renyi's*, volume 2, chapter On Measures of Entropy and Information Theory, pages 565–580. Akademia Kiado, Budapest, 1976.

[159] S. Roberts and R. Everson, editors. *Independent Component Analysis: Principles and Practice*. Cambridge University Press, 2001.

[160] W. Rudin. *Real and complex analysis*. McGraw-Hill, New York, 2nd edition, 1974.

[161] A. A. M. Saleh. Frequency-independent and frequency-dependent nonlinear models of twt amplifiers. *IEEE Trans. on Communications*, COM-29:1715–1720, 1981.

[162] M. Scarpiniti, R. Parisi, and A. Uncini. Flexible probability and comulative density functions estimation. *submitted to Electronics Letters*, 2009.

[163] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini. Blind separation of correlated complex signals using the generalized splitting function. Available online at

http://ispac.ing.uniroma1.it/scarpiniti/papers/U1.pdf, March 2007.

[164] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini. Generalized flexible splitting function outperforms classical approaches in blind signal separation of complex environment. In *Proc. of DSP2007*, pages 215–218, Cardiff, U.K., July 1-4 2007.

[165] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini. Generalized splitting functions for blind separation of complex signals. *Neurocomputing*, 71(10-12):2245–2270, June 2008.

[166] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini. *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, chapter Flexible Bllind Signal Separation in the Complex Domain, pages 284–323. IGI Global, Feb. 2009.

[167] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini. *New Directions in Neural Networks*, chapter A Flexible Natural Gradient Approach to Blind Separation of Complex Signals. Frontiers in Artificial Intelligence and Applications. IOS Pres, Amsterdam, March 2009.

[168] D. Schobben, K. Torkkola, and P. Smaragdis. Evaluation of blind signal separation methods. In *In Proc. of ICA and BSS*, pages 239–244, Aussois, France, Jenuary, 11-15 1999.

[169] I. J. Schoenberg. On variation diminishing approximation methods. *Journal of Inequality & Applications*, 12:249–274, 1959.

[170] O. Shalvi and E. Weinstein. New criteria for blind deconvolution of nonminimum phase systems (channels). *IEEE Trans. on Information Theory*, 36(2):312–321, 1990.

[171] O. Shalvi and E. Weinstein. Maximum likelihood and lower bounds in system identification with non-gaussian inputs. *IEEE Transaction on Information Theory*, 40(2):328–339, 1994.

[172] H. Silverman. *Complex variables*. Houghton Mifflin, Boston, 1975.

[173] V. P. Skitovitch. On a property of the normal distribution. *DAN SSSR*, 89:217–219, 1953.

[174] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. In *Proc. International workshop on Indipendence and Artificial Neural Networks*, Tenerife, Spain, February, 9-10 1998.

[175] M. Solazzi, F. Piazza, and A. Uncini. An adaptive spline nonlinear function for blind signal processing. In *Proc. of IEEE Whorkshop on neural networks for signal Processing*, volume X, pages 396–404, December 2000.

[176] M. Solazzi, F. Piazza, and A. Uncini. Nonlinear blind source separation by spline neural network. In *Proc.of ICASSP 2001*, volume 5, pages 2781–2784, Salt Lake city, USA, May, 8-11 2001.

[177] M. Solazzi and A. Uncini. Artificial neural networks with adaptive multidimensional spline activation functions. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 471–476vol.3, 24-27 July 2000.

[178] M. Solazzi and A. Uncini. Regularizing neural networks using flexible multivariate activation function. *Neural Networks*, 17:247–260, 2004.

[179] E. Sorouchyari. Blind separation of sources, part iii: Stabilty analysis. *Signal Processing*, 24:21–29, 1991.

[180] S. Squartini, F. Piazza, and A. Shawker. New riemannian metrics for improvement of convergence speed in ica based learning algorithms. In *Proc. IEEE International Symposium on Circuits and Systems ISCAS 2005*, pages 3603–3606 Vol. 4, 2005.

[181] M. Stinchcombe and H. White. Universal approximation using feedforward networks with nonsigmoid hidden layer activation functions. In *Proc. IJCNN*, pages I 613–617, Washington, D.C., 1989.

[182] M. Stinchcombe and H. White. Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights. In *Proc. IJCNN*, pages III 7–16, San Diego, CA, 1990.

[183] A. Taleb. A generic framework for blind source separation in structured nonlinear models. *IEEE Transaction on signal processing*, 50(8):1819–1830, August 2002.

[184] A. Taleb and C. Jutten. Nonlinear source separation: The post-nonlinear mixtures. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 279–284, Bruges, Belgium, 1997.

[185] A. Taleb and C. Jutten. On underdetermined source separation. In *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, volume 3, pages 1445–1448vol.3, 15-19 March 1999.

[186] A. Taleb and C. Jutten. Source separation in post nonlinear mixtures. *IEEE Transaction on Signal Processing*, 47(10):2807–2820, August 1999.

[187] A. Tarighat and A. H. Sayed. Least mean-phase adaptive filters with application to communications systems. *IEEE Signal Processing Letters*, 11(2):220–223, 2004.

[188] F. J. Theis. *Mathematics in Independent Component Analysis*. PhD thesis, Universitĺat Regensburg, 2002.

[189] F. J. Theis. A new concept for separability problems in blind source separation. *Neural Computation*, 16:1827–1850, 2004.

[190] F. J. Theis. Uniqueness of complex and multidimensional independent component analysis. *Signal Processing*, 84:951–956, 2004.

[191] F. J. Theis, C. Bauer, and E.W. Lang. Comparison of maximum entropy and minimal mutual information in a nonlinear setting. *Signal Processing*, 82:971–980, 2002.

[192] F. J. Theis, C. Bauer, C.G. Puntonet, and E.W. Lang. Pattern repulsion revisited. In *Proceeding of IWANN*, volume 2085 of *LNCS*, pages 778–786. Springer, 2001.

[193] F. J. Theis and P. Gruber. On model identifiability in analytic postnonlinear ica. *Neurocomputing*, 64:223–234, 2005.

[194] F. J. Theis and E.W. Lang. Maximum entropy and minimal mutual information in a nonlinear model. In *Proceeding of ICA*, pages 669–674, San Diego, USA, 2001.

[195] F. J. Theis and E.W. Lang. Uniqueness of linear independet component analysis and the skitovitch-darmois theorem. Preprint, available online at http://www.biologie.uni-regensburg.de/Biophysik/Theis/publications/preprints/theis03darmois.pdf, June 2003.

[196] P. Tichavský, Z. Koldovsky, and E. Oja. Corrections to "performance analysis of the fastica algorithm and cramér-rao bounds for linear independent component analysis". *IEEE Transaction on Signal Processing*, 56(4):1715–1716, April 2008.

[197] P. Tichavsky, Z. Koldovsky, and E. Oja. Asymptotic performance of the fastica algorithm for independent component analysis and its iprovements. In *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, pages 1084–1089, July 17-20 2005.

[198] P. Tichavsky, Z. Koldovsky, and E. Oja. Performance analysis of the fastica algorithm and cramèr-rao bounds for linear independent component analysis. *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, 54(4):1189–1203, April 2006.

[199] A. Uncini, F. Capparelli, and F. Piazza. Fast complex adaptive spline neural networks for digital signal processing. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 2, pages 903–908vol.2, 4-9 May 1998.

[200] A. Uncini and F. Piazza. Blind signal processing by complex domain adaptive spline neural network. *Transaction on Neural Networks*, 14(2):399–412, March 2003.

[201] A. Uncini, L. Vecci, P. Campolucci, and F. Piazza. Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization. *IEEE Trans. on Signal Processing*, 47(2):505–514, Feb. 1999.

[202] M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing. i. theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, Feb. 1993.

[203] N. Vakhania and N. Kandelaki. Random vectors with values in complex hilbert spaces. *Theory Probab. Appl.*, 41(1):116–131, Feb. 1996.

[204] M. Valkama, M. Renfors, and V. Koivunen. Advanced methods for i/q imbalance compensation in comunication receivers. *IEEE Transaction on Signal Processing*, 49(10):2335–2344, 2001.

[205] M. Valkama, M. Renfors, and V. Koivunen. Blind image suppression and carrier tracking in direct-conversion receivers based on i/q signal separation. In *Proc. 38th Asilomar Conf. Signal, Syst., Comput.*, volume 1, pages 603–606, Pacific Grove, CA, Nov. 2004.

[206] A. van den Bos. Complex gradient and hessian. In *IEE Proc. of Vision, Image and Signal Processing*, volume 141, pages 380–383, December 1994.

[207] L. Vecci, F. Piazza, and A. Uncini. Learning and approximation capabilities of adaptive spline activation function neural networks. *Neural Networks*, 11(2):259–270, March 1998.

[208] D. Vigliano, R. Parisi, and A. Uncini. An information theoretic approach to a novel nonlinear independent component analysis paradigm. *Signal Processing*, 85:997–1028, 2005.

[209] D. Vigliano, M. Scarpiniti, R. Parisi, and A. Uncini. A flexible blind source recovery in complex nonlinear environment. In *Proc. of IEEE International Symposium on Intelligent Control*, pages 3059–3063, Munich, Germany, October 4-6 2006.

[210] D. Vigliano, M. Scarpiniti, R. Parisi, and A. Uncini. Flexible ica approach to the nonlinear blind source separation in the complex domain. In *Proc. of EUSIPCO'06*, pages 2261–2265, Florence, Italy, September 4-8 2006.

[211] D. Vigliano, M. Scarpiniti, R. Parisi, and A. Uncini. Flexible ica in complex and nonlinear environment by mutual information minimization. In *Proc. of MLSP'06*, pages 59–63, Maynooth, Ireland, September 6-8 2006.

[212] D. Vigliano, M. Scarpiniti, R. Parisi, and A. Uncini. Flexible nonlinear blind signal separation in the complex domain. *International Journal of Neural System*, 18(2):105–122, April 2008.

[213] S. Visuri and V. Koivunen. Resolving ambiguities in subspace-based blind receiver for mimo channels. In *Proc. of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 589– 593, 3-6 Nov. 2002.

[214] F. Vitagliano, R. Parisi, and A. Uncini. Generalized splitting 2d flexible activation function. In *Lecture Notes in Computer Science*, volume 2859, pages 165–170, May 2003.

[215] T. P. von Hoff, A. G. Lindgren, and Kaelin A. N. Transpose properties in the stability and performance of the classic adaptive algorithms for bss and deconvolution. *Signal Processing*, 80(9):1807–1822, 2000.

[216] W. Wirtinger. Zur formalen theorie der funktionen vonmehr komplexen veranderlichen. *Mathematische Annalen*, 97(1):357–375, 1927.

[217] D. Xu, J. C. Principe, J. Fisher III, and H.-C. Wu. A novel measure for independent component analysis (ica). In *Proceeding of ICASSP'98*, volume 2, pages 1161–1164, 1998.

[218] H. H. Yang and S-I. Amari. Adaptive on-line learning algorithms for blind separation: Maximum entropy and minimum mutual information. *Neural Computation*, 9:1457–1482, 1997.

[219] H. Y. Yang, S.I. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in nonlinear mixture. *Signal Processing*, 64:291–300, 1998.

[220] D. Yellin and B. Friedlander. Multichannel system identification and deconvolution: Performance bounds. *IEEE Transaction on Signal Processing*, 47(5):1410–1414, May 1999.

[221] F. Zhao, T. Jin, P. Wang, X. Hu, and S.-G. Kim. Sources of phase changes in bold and cbv-weighted fmri. *Mag. Res. Med.*, 57(3):520–527, 2007.

A

Elements of Information Theory

—Beyond each corner new directions lie in wait.
Stanislaw Lec

In this appendix we introduce some elements of the information theory, which have been used in this thesis. In particular we must define the entropy of a random variable, the joint and conditional entropy of several random variables, the Kullback-Leibler divergence and the mutual information ([144], [55], [105]).

In fact when the neural weights elaborate the inputs, the information transported by these will be maximized. Clearly, the information theory (IT) is one of the powerful formal instrument of the neural networks theory.

A.1 Entropy

Let X be a random discrete variable, say

$$X = \{x_k \,|k = 0, \pm 1, \ldots, \pm K\}$$

where we have $2K + 1$ levels. We define the probability of the event $X = x_k$ as follows

$$p_k = P\{X = x_k\}$$

where obviously $0 \leq p_k \leq 1$ and $\sum_{k=-K}^{K} p_k = 1$.

If the event $X = x_k$ has probability equal to 1, we have no surprise and so there is not any information, transported by the X random variable. On the contrary, if $p_k < 1$ we have some uncertainty and so some information. Thus, the information is inversely proportional to the probability of the event.

Definition 29 *If the event $X = x_k$ has probability p_k, the following quantity is defined as* ***information gain****:*

$$I(x_k) = \ln\left(\frac{1}{p_k}\right) = -\ln p_k \tag{A.1}$$

For the natural logarithm this quantity is measured in *nat*, while for logarithm with base 2 is measured in *bit*.

The information gain has the following three properties:

1. $I(x_k) = 0$ for $p_k = 1$;
2. $I(x_k) > 0$ for $0 \leq p_k < 1$;
3. $I(x_k) > I(x_j)$ for $p_k < p_j$.

The less probable event takes more information.

Definition 30 *The entropy $H(X)$ of the random variable X is the mean value of the information gain $I(x_k)$:*

$$H(X) = E\{I(x_k)\} = \sum_{k=-K}^{K} p_k I(x_k) = -\sum_{k=-K}^{K} p_k \ln p_k \tag{A.2}$$

The discrete entropy has the following properties:

1. $0 \leq H(X) \leq 2K + 1$;
2. $H(X) = 0$ if and only if $p_k = 1, \forall k$;
3. $H(X) = \log_2(2K + 1)$ if and only if $p_k = 1/(2K + 1), \forall k$.

From the third property , the entropy is maximum if we have an uniform distribution.

Lemma 10 *For two distributions p_k and q_k of a random variable, we have that the following quantity*

$$\sum_k p_k \ln\left(\frac{p_k}{q_k}\right) \geq 0 \tag{A.3}$$

is equal to zero if and only if $p_k = q_k$, for every k.

We can extend these concepts for the random continuous variables:

Definition 31 *The differential entropy $h(X)$ of a random continuous variable with probability density function (pdf) $f_X(x)$, is defined as*

$$h(X) = -\int_{-\infty}^{\infty} f_X(x) \ln f_X(x) dx = -E\{\ln f_X(x)\} \tag{A.4}$$

This definition of the differential entropy can be derived from the entropy definition as a limit case. We can assume that $x_k = k\delta_x$ and that $\delta_x \to 0$, thus X is constant in $[x_k, x_k + \delta_x]$ with probability $f_X(x)\delta_x$. Then the entropy of X can be written as

$$\begin{aligned}
&H(X) = -\lim_{\delta_x} \sum_{k=-\infty}^{\infty} f_X(x_k)\delta_x \ln[f_X(x)\delta_x] = \\
&= \lim_{\delta_x \to 0} \left[\sum_{k=-\infty}^{\infty} f_X(x_k) \left[\ln f_X(x_k)\right] \delta_x + \sum_{k=-\infty}^{\infty} f_X(x_k)\delta_x \right] = \\
&= -\int_{-\infty}^{\infty} f_X(x) \ln f_X(x) dx - \lim_{\delta_x \to 0} \ln \delta_x \int_{-\infty}^{\infty} f_X(x) dx = \\
&= h(X) - \lim_{\delta_x \to 0} \ln \delta_x \to h(X)
\end{aligned}$$

remembering that $\int_{-\infty}^{\infty} f_X(x)dx = 1$. The problem for the term $ln\delta_x$ can be eliminated by the concept of the differential entropy, where the term $-ln\delta_x$ is assumed as a reference term.

The differential entropy has the following properties:

1. $h(X) = h(X + c)$, where c is a constant;
2. $h(aX) = h(X) + ln|a|$, where a is a scaling factor;
3. $h(\mathbf{A}X) = h(X) + ln|det\mathbf{A}|$, for vector random variables.

For a uniform random variable we have the following

Lemma 11 *The differential entropy of a uniform random variable X in the interval $[0, a]$, is $h(X) = lna$*

Proof. Directly we have

$$h(X) = -\int_{-\infty}^{\infty} \frac{1}{a} \ln \frac{1}{a} dx = -\frac{1}{a} \ln \frac{1}{a} \int_0^a dx = \ln a$$

■

If we have a multidimensional random vector $\mathbf{X} = [x_1, x_2, \ldots, x_N]$, we can give the

Definition 32 *If we have a multidimensional random vector $\mathbf{X}$, we define the joint entropy as the following expression:*

$$H(\mathbf{X}) = -\int_{x_1} \int_{x_2} \ldots \int_{x_N} f_X(x_1, x_2, \ldots, x_N) \ln f_X(x_1, x_2, \ldots, x_N) dx_1 \cdot dx_2 \cdot \ldots \cdot dx_N \tag{A.5}$$

The joint entropy represents the mean information transported by the N random variables.

A.2 Kullback-Leibler divergence (KLD)

An important distance measure between two density probability functions is the Kullback-Leibler divergence, defined from the following [55]

Definition 33 *Let $f_X(x)$ and $g_X(x)$ be the density probability functions (pdf) of the multidimensional random variable* $\mathbf{X}$*, then the following quantity is the* ***Kullback-Leibler divergence*** *(KLD):*

$$D_{f_X \| g_X} = \int_{-\infty}^{\infty} f_X(x) \ln \left(\frac{f_X(x)}{g_X(x)} \right) dx \tag{A.6}$$

The Kullback-Leibler divergence has the following properties:

1. $D_{f_X \| g_X} \geq 0$, and is zero if and only if $f_X(x) = g_X(x)$;
2. The $D_{f_X \| g_X}$ is invariant for the following transformation on X:
   (a) Permutation of the elements of X;
   (b) Scaling;
   (c) Non linear and monotonic transformation.

Now we have to introduce the concept of marginal random variable:

Definition 34 *The i-th marginal pdf of the X_i element of* $\mathbf{X}$*, is defined as*

$$\tilde{f}_{x_i}(x_i) = \int_{-\infty}^{\infty} f_X(x) dx^{(i)}, i = 1, 2, \ldots, m \tag{A.7}$$

where $x^{(i)}$ is the vector without the i-th element X_i.

Now we can introduce the Kullback-Leibler divergence between the pdf $f_X(x)$ and the product of its marginal distributions $\prod_{i=1}^{m} \tilde{f}_{x_i}(x_i)$, defined as

$$D_{f_X \| \tilde{f}_X} = \int_{-\infty}^{\infty} f_X(\mathbf{x}) \ln \left(\frac{f_X(\mathbf{x})}{\prod_{i=1}^{m} f_{X_i}(x_i)} \right) d\mathbf{x} \tag{A.8}$$

This expression can be rewritten as

$$D_{f_X \| \tilde{f}_X} = \int_{-\infty}^{\infty} f_X(\mathbf{x}) \ln f_X(\mathbf{x}) d\mathbf{x} - \sum_{i=1}^{m} \int_{-\infty}^{\infty} f_X(\mathbf{x}) \ln \tilde{f}_{X_i}(x_i) d\mathbf{x}$$

In the second term we can pose $d\mathbf{x} = d\mathbf{x}^{(i)}dx_i$, so we have

$$\int_{-\infty}^{\infty} f_X(\mathbf{x}) \ln \tilde{f}_{X_i}(x_i) d\mathbf{x} = \int_{-\infty}^{\infty} \ln \tilde{f}_{X_i}(x_i) \int_{-\infty}^{\infty} f_X(\mathbf{x}) d\mathbf{x}^{(i)} dx_i$$

But the internal integral is the marginal distribution of X_i, so we have

$$\int_{-\infty}^{\infty} f_X(\mathbf{x}) \ln \tilde{f}_{X_i}(x_i) d\mathbf{x} = \int_{-\infty}^{\infty} \tilde{f}_{X_i}(x_i) \ln \tilde{f}_{X_i}(x_i) dx_i = -\tilde{h}(X_i), i = 1, 2, \ldots, m$$

where $h(\tilde{X}_i)$ is the i-th differential and marginal entropy.

Finally we obtain the following expression for the DKL:

$$D_{f_X \| \tilde{f}_X} = -h(\mathbf{X}) + \sum_{i=1}^{m} \tilde{h}(X_i) \tag{A.9}$$

This expression is very important in many applications, such as blind signal separation.

A.3 Mutual Information

Let us consider a system with X and Y the input random variable and the output random variable respectively. $H(X)$ is the measure of the uncertainty on the input X. We would know the uncertainty on input X if we have the observation Y. We have to give the following

Definition 35 *The conditional entropy $H(X|Y)$ of the random variable X given the random variable Y, is expressed as*

$$H(X\,|Y) = H(X,Y) - H(Y) \tag{A.10}$$

The conditional entropy represents the amount of uncertainty on X after we have observed the output Y. It has the property:

$$0 \leq H\,(X\,|Y\,) \leq H(X)$$

Hence, we can deduce that $H(X) - H\,(X\,|Y\,)$ represents the uncertainty on the input X given by the observation Y. This latter quantity is described as

Definition 36 *We define the mutual information $I(X,Y)$ between the two discrete random variables X and Y the following quantity:*

$$I(X;Y) = H(X) - H\,(X\,|Y\,) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \ln \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{A.11}$$

The entropy is the mutual information when $Y = X$: $H(X) = I(X;X)$.

The mutual information has the following property:

1. $I(X;Y) = I(Y;X)$, it is symmetric;
2. $I(X;Y) \geq 0$ it is non-negative; it is zero if and only if X and Y are statistically independent;
3. $I(X;Y) = H(Y) - H(Y|X)$, it is reciprocal.

The relation between the mutual information $I(X;Y)$ and the entropies $H(X)$, $H(Y)$ and $H(X,Y)$ is shown in the following figure

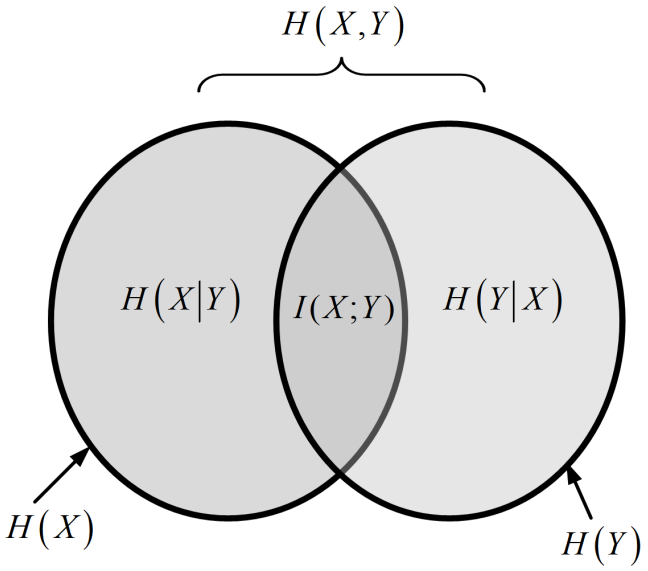

Fig. A.1: Relation between $I(X;Y)$, $H(X)$, $H(Y)$ and $H(X,Y)$

We can extend the concept of mutual information for continuous random variable, as follows

Definition 37 *We define the mutual information $I(X,Y)$ between the two continuous random variables X and Y the following quantity:*

$$I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y) \ln\left(\frac{f_X(x|y)}{f_X(x)}\right) dxdy \tag{A.12}$$

where $f_{X,Y}(x,y)$ is the joint pdf of X and Y, while $f_X(x|y$ is the conditioned pdf of X given Y.

Similarly the mutual information for continuous random variable has the following properties:

1. $I(X;Y) = h(X) - h(X\,|Y) = h(Y) - h(Y\,|X) = I(Y;X)$, it is symmetric and reciprocal;

2. $I(X;Y) \geq 0$, it is non-negative and it is zero if, and only if X and Y are statistically independent.

The differential conditioned entropy can be calculated as follows:

$$h\,(X\,|Y\,) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(x,y)\ln f_X(x\,|y)dxdy \tag{A.13}$$

Now noting that $f_{X,Y}(x,y) = f_Y(y\,|x)f_X(x)$ and remembering the expression of the Kullback-Leibler divergence, we can write

$$I(\mathbf{X};\mathbf{Y}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_{X,Y}(\mathbf{x},\mathbf{y})\ln\left(\frac{f_{X,Y}(\mathbf{x},\mathbf{y})}{f_X(\mathbf{x})f_Y(\mathbf{y})}\right)d\mathbf{x}d\mathbf{y} = D_{f_{X,Y}\|f_X\cdot f_Y} \tag{A.14}$$

So the mutual information corresponds to the Kullback-Leibler divergence between the joint pdf $f_{X,Y}(x,y)$ and the product of its marginal pdf's $f_X(x)$ and $f_Y(y)$.

We give two important and useful theorem [144]:

Theorem 22 *Given an n-dimensional gaussian random variable $\mathbf{X}$, with mean μ and covariance matrix $\mathbf{K}$, such that*

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^n|\mathbf{K}|^{1/2}}e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\mathbf{K}^{-1}(\mathbf{x}-\mu)}$$

then we have that

$$H(\mathbf{x}) = \frac{1}{2}\ln\left[(2\pi e)^n\,|\mathbf{K}|\right] \tag{A.15}$$

Theorem 23 *The entropy of a n-dimensional random variable $\mathbf{X}$ is smaller than the entropy of a n-dimensional gaussian random variable with equal mean and covariance matrix:*

$$H(\mathbf{x}) \leq \frac{1}{2}\ln\left[(2\pi e)^n\,|\mathbf{K}|\right] \tag{A.16}$$

We remember that the covariance matrix $\mathbf{K}$ is defined as $\mathbf{K} = E\left\{\mathbf{x}\mathbf{x}^T\right\}$.

A.4 Negentropy

The negentropy can be seen as a very important measure of nongaussianity, and it is described as

Definition 38 *The **negentropy** $J(\mathbf{X})$ of a random variable $\mathbf{X}$ is*

$$J(X) = H(\mathbf{X}_{gauss}) - H(\mathbf{X}) \tag{A.17}$$

where $\mathbf{X}_{gauss}$ is a Gaussian random variable of the same covariance matrix as $\mathbf{X}$.

The negentropy has the following properties:

1. $J(\mathbf{X}) \geq 0$, it is zero if and only if $\mathbf{X}$ has a Gaussian distribution;
2. $J(\mathbf{MX}) = J(\mathbf{X})$, it is invariant for any invertible linear transformations.

Proof. Using the relation (A.15) we obtain:

$$\begin{gathered} J(\mathbf{Mx}) = \tfrac{1}{2}\ln\left|\det\left(\mathbf{MKM}^T\right)\right| + \tfrac{n}{2}\left[1+\ln 2\pi\right] - \left(H(\mathbf{x}) + \ln|\det\mathbf{M}|\right) = \\ = \tfrac{1}{2}\ln|\det\mathbf{K}| + 2\tfrac{1}{2}\ln|\det\mathbf{M}| + \tfrac{n}{2}\left[1+\ln 2\pi\right] - H(\mathbf{x}) - \ln|\det\mathbf{M}| = \\ = \tfrac{1}{2}\ln|\det\mathbf{K}| + \tfrac{n}{2}\left[1+\ln 2\pi\right] - H(\mathbf{x}) = \\ = H(\mathbf{x}_{gauss}) - H(\mathbf{x}) = J(\mathbf{x}) \end{gathered}$$

■

The negentropy is the optimal estimator of nongaussianity. The problem in using negentropy is that it is computationally very difficult. Estimating negentropy using the definition would require an estimate of the pdf. Therefore, some approximations have to be used.
The classical method of approximating negentropy is using higher-order moments as follows:

$$J(y) \approx \frac{1}{12}E\left\{y^3\right\}^2 + \frac{1}{48}kurt(y)^2 \tag{A.18}$$

The random variable $\mathbf{y}$ is assumed to be of zero mean and unit variance. However, the validity of these approximations may be rather limited. In particular, these approximations suffers from the non robustness encountered with kurtosis.

B

Elements of Complex Variables

—Climb mountains to see lowlands.
Old Asian Proverb

In this appendix we introduce some elements of the complex variables and functions, which have been used in this thesis. In particular we have to define some useful statistical properties of complex random variables (r.v.s). A good text for complex variables can be found in [160, 172, 157, 114], while most of the results of this appendix are drawn from [66].

B.1 Complex variables and vectors

Let we pose $z = z_R + jz_I$ a complex variable ($z \in \mathbb{C}$), where z_R, z_I are the real and imaginary parts of the complex variable z rispectively and $j = \sqrt{-1}$ is the *imaginary unit*.

We denote with $z^* = z_R - jz_I$ the complex *conjugate* of z, while its *modulus* is denoted as $|z| = \sqrt{zz^*} = \sqrt{{z_R}^2 + {z_I}^2}$.

This representation of complex numbers is called *Cartesian form*; there exists another representation known as *polar form*: $z = me^{j\vartheta}$, where $m > 0, \vartheta \in \mathbb{R}$. The number m coincides the modulus of the complex variable z while the number ϑ is called an *argument* of z; the argument $\vartheta = Arg(z)$ such that $-\pi \leq \vartheta < \pi$ is called the *principal argument*. We can pass through the Cartesian form to the polar form using the following relations:

$$\begin{aligned} m &= \sqrt{z_R^2 + z_I^2} \\ \vartheta &= \operatorname{atan}\tfrac{z_I}{z_R} \end{aligned} \tag{B.1}$$

and vice versa from polar representation to the Cartesian one using:

$$\begin{aligned} z_R &= m\cos\vartheta \\ z_I &= m\sin\vartheta \end{aligned} \tag{B.2}$$

The real part of a complex variable can be obtained as $z_R = Re\{z\}$ while the imaginary part as $z_I = Im\{z\}$.

A vector of complex variables z_k ($k = 1, \ldots, N$) is denoted by the bold face lower case letter $\mathbf{z} = (z_1, z_2, \ldots, z_N)^T$, where T is the ordinary *transpose* operator. The real part of a N-dimensional complex vector $\mathbf{z}$ is denoted by $\mathbf{z}_R = Re\{\mathbf{z}\}$ and similar for the imaginary part $\mathbf{z}_I = Im\{\mathbf{z}\}$. The *Euclidean norm* of the vector $\mathbf{z}$ is denoted by $\|\mathbf{z}\|^2 = \langle \mathbf{z}, \mathbf{z} \rangle = \mathbf{z}\mathbf{z}^H$, where $\langle \cdot, \cdot \rangle$ is the inner product and the superscript H denotes the conjugate transpose.

A complex matrix $\mathbf{C} \in \mathbb{C}^{N \times N}$ is termed *symmetric* if $\mathbf{C}^T = \mathbf{C}$ and *Hermitian* if $\mathbf{C}^H = \mathbf{C}$. Furthermore, the matrix $\mathbf{C}$ is *orthogonal* if $\mathbf{C}^T\mathbf{C} = \mathbf{C}\mathbf{C}^T = \mathbf{I}$ and *unitary* if $\mathbf{C}^H\mathbf{C} = \mathbf{C}\mathbf{C}^H = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix.

B.2 The Augmented representation

Let $\mathbf{C} = \mathbf{C}_R + j\mathbf{C}_I \in \mathbb{C}^{m \times n}$ and $\mathbf{z} = \mathbf{z}_R + j\mathbf{z}_I \in \mathbb{C}^n$. We use the following notation

$$\mathbf{C}_{\mathbb{R}} = \begin{pmatrix} \mathbf{C}_R & -\mathbf{C}_I \\ \mathbf{C}_I & \mathbf{C}_R \end{pmatrix} \quad \text{and} \quad \mathbf{z}_{\mathbb{R}} = \begin{pmatrix} z_R \\ z_I \end{pmatrix} \tag{B.3}$$

for the associated $2m \times 2n$ real matrix and $2n$-variate real vector, respectively. The mapping $\mathbf{z} \mapsto \mathbf{z}_{\mathbb{R}}$ gives naturally a group isomorphism between the additive Abelian groups $\mathbb{C}^n$ and $\mathbb{R}^{2n}$.In the case $m = n = 1$, the mapping given by $\mathbf{C} \mapsto \mathbf{C}_{\mathbb{R}}$ defines a field isomorphism between the complex numbers and a subset of real two dimensional matrices.

Now consider the mapping

$$\mathbf{C}\mathbf{z} \mapsto (\mathbf{C}\mathbf{z})_{\mathbb{R}} = \mathbf{C}_{\mathbb{R}}\mathbf{z}_{\mathbb{R}} \tag{B.4}$$

it is continuous and therefore preserves the topological properties, i.e., it is a homeomorphism. Let $diag(\mathbf{z})$ denote the *diagonal matrix* with components of $\mathbf{z}$ in its main diagonal and zeros elsewhere. Since $\mathbb{C}^n$ is a vector space, where the scalar multiplication for $c \in \mathbb{C}$ is given by

$$c\mathbf{z} \triangleq \begin{pmatrix} cz_1 \\ \vdots \\ cz_n \end{pmatrix} = diag\left(\begin{pmatrix} c & \cdots & c \end{pmatrix}\right)\mathbf{z} \tag{B.5}$$

the mapping (B.4) defines a vector space isomorphism between the standard n-dimensional complex vector space and a $2n$-dimensional real-valued vector space given by the mapping. It is important to realize that this associated

real-valued vector space is *not* isomorphic to the standard real vector space $\mathbb{R}^{2n}$. Furthermore, by equating ${\mathbf{z}_1}^H$ with $\mathbf{C}$ in (B.4) it is easily verified that the mapping $\mathbb{C} \to \mathbb{R}^2 : \mathbf{z}_1^H \mathbf{z}_2 \mapsto \left(\mathbf{z}_1^H\right)_{\mathbb{R}} (\mathbf{z}_2)_{\mathbb{R}}$ associates a (complex) inner product for $\mathbb{R}^{2n}$. Therefore, the mapping (B.4) is also a *Hilbert space isomorphism*. Again, it should be emphasized that the inner product given by the mapping is not the standard Euclidean inner product in $\mathbb{R}^{2n}$. However, the vector norms, and hence metrics, are equivalent in both.

The following properties are easily established.

Lemma 12 *Let $\mathbf{C} \in \mathbb{C}^{n\times n}$ and $\mathbf{z} \in \mathbb{C}^n$, then*

1. $|det(\mathbf{C})|^2 = det(\mathbf{C}_{\mathbb{R}})$;
2. $\mathbf{C}$ *is Hermitian iff* $\mathbf{C}_{\mathbb{R}}$ *is symmetric. Then* $det(\mathbf{C})^2 = det(\mathbf{C}_{\mathbb{R}})$ *and* $2 \times rank(\mathbf{C}) = rank(\mathbf{C}_{\mathbb{R}})$;
3. $\mathbf{C}$ *is nonsingular iff* $\mathbf{C}_{\mathbb{R}}$ *is nonsingular;*
4. $\mathbf{C}$ *is unitary iff* $\mathbf{C}_{\mathbb{R}}$ *is orthogonal;*
5. $\mathbf{z}^H \mathbf{C} \mathbf{z} = {\mathbf{z}_{\mathbb{R}}}^T \mathbf{C}_{\mathbb{R}} \mathbf{z}_{\mathbb{R}}$;
6. $\mathbf{C}$ *is Hermitian positive definite iff* $\mathbf{C}_{\mathbb{R}}$ *is symmetric positive definitive;*
7. *any polynomial with complex coefficients in variables* $\mathbf{z}_{\mathbb{R}}$ *can be equivalently given in variables* $(\mathbf{z}, \mathbf{z}^*)$.

Proof. These properties are direct consequences of the isomorphism. The last property follows from the identities $\mathbf{z}_{\mathbb{R}} = \frac{1}{2}(\mathbf{z} + \mathbf{z}^*)$ and $\mathbf{z}_{\mathbb{I}} = \frac{-j}{2}(\mathbf{z} - \mathbf{z}^*)$. ■

Since the variables $(\mathbf{z}, \mathbf{z}^*)$ in Lemma 12, 7) are dependent, we call such complex polynomials *wide sense polynomials*. The idea of using also the complex conjugate variable has turned out to be highly useful in.

B.3 Complex functions

Let $f(z)$ be a complex function, where $z = z_R + jz_I \in \mathbb{C}$, then it is possible to write

$$f(z) = u(z_R, z_I) + jv(z_R, z_I) \tag{B.6}$$

where $u(z_R, z_I)$ is the real part function and $v(z_R, z_I)$ is the imaginary part function.

The fundamental result for the differentiability of the complex-valued function (B.6) is given by the *Cauchy-Riemann* equations:

$$\begin{array}{l} \frac{\partial u}{\partial z_R} = \frac{\partial v}{\partial z_I}, \\ \frac{\partial v}{\partial z_R} = -\frac{\partial u}{\partial z_I} \end{array} \tag{B.7}$$

which summarize the conditions for the derivative to assume the same value regardless of the direction of approach when $\Delta z \to 0$. These conditions, when considered carefully, make it clear that the definition of complex differentiability is quite stringent and imposes a strong structure on $u(z_R, z_I)$ and $v(z_R, z_I)$, the real and imaginary parts of the function, and consequently on $f(z)$. Also, obviously most cost (objective) functions do not satisfy the Cauchy-Riemann equations as these functions are typically $f : \mathbb{C} \to \mathbb{R}$ and thus have $v(z_R, z_I) = 0$.

An elegant approach due to Wirtinger [216, 114] relaxes this strong requirement for differentiability, and defines a less stringent form for the complex domain. More importantly, it describes how this new definition can be used for defining complex differential operators that allow computation of derivatives in a very straightforward manner in the complex domain, by simply using real differentiation results and procedures.

In the development, the commonly used definition of differentiability that leads to the Cauchy-Riemann equations is identified as *complex differentiability* and functions that satisfy the condition on a specified open set as *complex analytic* (or *complex holomorphic*). A fundamental result for the analytic functions is the following

Theorem 24 (Liouville) *If $f(z)$ is entire and bounded on the complex plane $\mathbb{C}$, then $f(z)$ is a constant function.*

The more flexible form of differentiability is identified as *real differentiability*, and a function is called real differentiable when $u(z_R, z_I)$ and $v(z_R, z_I)$ are differentiable as functions of real-valued variables z_R and z_I. Then, one can write the two real-variables as $z_R = (z + z^*)/2$ and $z_I = -j(z - z^*)/2$, and use the chain rule to derive the operators for differentiation given in the theorem below. The key point in the derivation is regarding the two variables z and z^* as independent from each other, which is also the main trick that allows us to make use of the elegance of Wirtinger calculus. Hence, we consider a given function $f : \mathbb{C} \to \mathbb{C}$ as $f : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ by writing it as $f(z) = f(z_R, z_I)$, and make use of the underlying $\mathbb{R}^2$ structure. The main result in this context is stated by Brandwood as follows [22].

Theorem 25 *Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ be a function of real variables z_R and z_I such that $g(z, z^*) = f(z_R, z_I)$, where $z = z_R + jz_I$ and that g is analytic with respect to z and z^* independently. Then,*

1. *the partial derivatives*

$$\begin{aligned} \frac{\partial g}{\partial z} &= \frac{1}{2}\left(\frac{\partial f}{\partial z_R} - j\frac{\partial f}{\partial z_I}\right), \\ \frac{\partial g}{\partial z^*} &= \frac{1}{2}\left(\frac{\partial f}{\partial z_R} + j\frac{\partial f}{\partial z_I}\right) \end{aligned} \tag{B.8}$$

can be computed by treating z^* as a constant in g and z as a constant, respectively;

2. *a necessary and sufficient condition for f to have a stationary point is that $\partial g/\partial z = 0$. Similarly, $\partial g/\partial z = 0$ is also a necessary and sufficient condition.*

Therefore, when evaluating the gradient, we can directly compute the derivatives with respect to the complex argument, rather than calculating individual real-valued gradients as typically performed in the literature (see, e.g., [114]). The requirement for the analyticity of $g(z, z^*)$ with respect to z and z^* is independently equivalent to the condition on real differentiability of $f(z_R, z_I)$ since we can move from one form of the function to the other using the simple linear transformation given above [157]. When $f(z)$ is complex analytic, that is, when the Cauchy-Riemann conditions hold, $g(\cdot)$ becomes a function of only z, and the two derivatives, the one given in the theorem and the traditional one coincide.

The case we are typically interested in the development of signal processing algorithms is given by $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and is a special case of the result stated in the theorem. Hence we can employ the same procedure—taking derivatives independently with respect to z and z^*, in the optimization of a real-valued function as well. In the rest, we consider such functions from the general $f : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ case for completeness.

Example 10.

As a simple example, consider the function $g(z, z^*) = zz^* = |z|^2 = {z_R}^2 + {z_I}^2 = f(z_R, z_I)$. We have $(1/2)(\partial f/\partial z_R + j(\partial f/\partial z_I)) = z_R + jz_I = z$, which we can also evaluate as $\partial g/\partial z^* = z$, that is, by treating z as a constant in g when calculating the partial derivative.

The complex gradient defined by Brandwood [22] has been extended by van den Bos [206] to define a complex gradient and Hessian in $\mathbb{C}^{2N}$ by defining a mapping

$$\mathbf{z} \in \mathbb{C}^N \mapsto \tilde{\mathbf{z}} = \begin{bmatrix} z_1 \\ z_1^* \\ \vdots \\ z_N \\ z_N^* \end{bmatrix} \in \mathbb{C}^{2N} \tag{B.9}$$

Note that the mapping allows a direct extension of Wirtinger's result to the multidimensional space through N mappings of the form $(z_{R,k}, z_{I,k}) \mapsto (z_k, {z_k}^*)$, where $z = z_R + jz_I$, so that one can make use of Wirtinger derivatives. Since the transformation from $\mathbb{R}^2$ to $\mathbb{C}^2$ is a simple linear invertible mapping, one can work in either space, depending on the convenience offered by each. In [206], it is shown that such a transformation allows the

definition of a Hessian, hence of a Taylor series expansion very similar to the one in the real case, and the Hessian matrix $\mathbf{H}$ defined in this manner is naturally linked to the complex $\mathbb{C}^{N\times N}$ Hessian $\mathbf{G}$ in that if λ is an eigenvalue of $\mathbf{G}$, then 2λ is the corresponding eigenvalue of $\mathbf{H}$. The result implies that the positivity of the eigenvalues as well as the conditioning of the Hessian matrices are shared properties of the two matrices, that is, of the two representations.

B.3.1 Optimization in the Complex Domain

Vector case

We define $\langle \cdot, \cdot \rangle$ as the scalar inner product between two matrices $\mathbf{W}$ and $\mathbf{V}$ as

$$\langle \mathbf{W}, \mathbf{V} \rangle = \mathrm{Trace}(\mathbf{V}^{\mathrm{H}}\mathbf{W}) \tag{B.10}$$

so that $\langle \mathbf{W}, \mathbf{W} \rangle = \|\mathbf{W}\|^2_{\mathrm{Fro}}$, where the subscript Fro denotes the Frobenius norm. For vectors, the definition simplifies to $\langle \mathbf{w}, \mathbf{v} \rangle = \mathbf{v}^H\mathbf{w}$.

We define the gradient vector $\nabla_{\mathbf{z}} = [\partial/\partial z_1, \partial/\partial z_2, \ldots, \partial/\partial z_N]^T$ for vector $z = [z_1, z_2, \ldots, zN]^T$ with $z_k = z_{R,k} + jz_{I,k}$ in order to write the first-order Taylor series expansion for a function $g(z, z^*) : \mathbb{C}^N \times \mathbb{C}^N \to \mathbb{R}$,

$$\Delta g = \langle \Delta\mathbf{z}, \nabla_{\mathbf{z}^*} g \rangle + \langle \Delta\mathbf{z}^*, \nabla_{\mathbf{z}} g \rangle = 2\mathrm{Re}\left\{ \langle \Delta\mathbf{z}, \nabla_{\mathbf{z}^*} g \rangle \right\} \tag{B.11}$$

where the last equality follows because $g(\cdot, \cdot)$ is real valued.

Using the Cauchy-Schwarz-Bunyakovski inequality [128], it is straightforward to show that the first-order change in $g(\cdot, \cdot)$ will be maximized when $\Delta\mathbf{z}$ and the gradient $\nabla_{\mathbf{z}^*} g$ are collinear. Hence, it is the gradient with respect to the conjugate of the variable, $\nabla_{\mathbf{z}^*} g$, that defines the direction of the maximum rate of change in $g(\cdot, \cdot)$ with respect to $\mathbf{z}$, not $\nabla_{\mathbf{z}} g$ as sometimes noted in the literature. Thus the gradient optimization of $g(\cdot, \cdot)$ should use the update

$$\Delta\mathbf{z} = \mathbf{z}_{t+1} - \mathbf{z}_t = -\mu\nabla_{\mathbf{z}^*} g \tag{B.12}$$

as this form leads to a nonpositive increment given by $\Delta g = -2\mu \|\nabla_{\mathbf{z}^*} g\|^2$, while the update using $\Delta\mathbf{z} = -\mu\nabla_{\mathbf{z}} g$ results in updates $\Delta g = -2\mu\mathrm{Re}\left\{ \langle \nabla_{\mathbf{z}^*} g, \nabla_{\mathbf{z}} g \rangle \right\}$, which are not guaranteed to be nonpositive.

Based on (B.11), similar to a scalar function of two real vectors, the second-order Taylor series expansion of $g(\mathbf{z}, \mathbf{z}^*)$ can be written as

$$\Delta^2 g = \frac{1}{2}\left\langle \frac{\partial g}{\partial\mathbf{z}\partial\mathbf{z}^T}\Delta\mathbf{z}, \Delta\mathbf{z}^* \right\rangle + \frac{1}{2}\left\langle \frac{\partial g}{\partial\mathbf{z}^*\partial\mathbf{z}^H}\Delta\mathbf{z}^*, \Delta\mathbf{z} \right\rangle + \left\langle \frac{\partial g}{\partial\mathbf{z}\partial\mathbf{z}^H}\Delta\mathbf{z}^*, \Delta\mathbf{z}^* \right\rangle \tag{B.13}$$

Next, we derive the same complex gradient update rule using another approach, which provides the connection between the real and complex domains. We first introduce the following fundamental mappings that are similar in nature to those introduced in [206].

Proposition 6 *Given a function $g(\mathbf{z}, \mathbf{z}^*) : \mathbb{C}^N \times \mathbb{C}^N \to \mathbb{R}$ that is real differentiable and $f : \mathbb{R}^{2N} \to \mathbb{R}$ such that $g(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{w})$, where $\mathbf{z} = [z_1, z_2, \ldots, z_N]^T$, $\mathbf{w} = [z_{R,1}, z_{I,1}, z_{R,2}, z_{I,2}, \ldots, z_{R,N}, z_{I,N}]^T$, and $z_k = z_{R,k} + j z_{I,k}$, $k \in \{1, 2, \ldots, N\}$, then*

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{w}} &= \mathbf{U}^H \frac{\partial g}{\partial \bar{\mathbf{z}}^*} \\ \frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} &= \mathbf{U}^H \frac{\partial g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \end{aligned} \tag{B.14}$$

where $\mathbf{U}$ is defined by $\bar{\mathbf{z}} \triangleq \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix}$ and satisfies $U^{-1} = (1/2)\mathbf{U}^H$.

In the following two propositions, we show how to use the same mappings we defined above to obtain first- and second-order derivatives, and hence algorithms, in $\mathbb{C}^N$ in an efficient manner.

Proposition 7 *Given functions g and f defined as in Proposition 6, one has the complex gradient update rule*

$$\Delta \mathbf{z} = -2\mu \frac{\partial g}{\partial \mathbf{z}^*} \tag{B.15}$$

which is equivalent to the real gradient update rule

$$\Delta \mathbf{w} = -\mu \frac{\partial f}{\partial \mathbf{w}} \tag{B.16}$$

where $\mathbf{z}$ and $\mathbf{w}$ are as defined in Proposition 6 as well.

Proposition 8 *Given functions g and f defined as in Proposition 6, one has the complex Newton update rule*

$$\Delta \mathbf{z} = -\left(\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1\right)^{-1} \left(\frac{\partial g}{\partial \mathbf{z}^*} - \mathbf{H}_1^* \mathbf{H}_2^{-1} \frac{\partial g}{\partial \mathbf{z}}\right) \tag{B.17}$$

which is equivalent to the real Newton update rule

$$\frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} \Delta \mathbf{w} = -\frac{\partial f}{\partial \mathbf{w}} \tag{B.18}$$

where

$$\mathbf{H}_1 = \frac{\partial^2 f}{\partial \mathbf{z} \partial \mathbf{z}^T}, \quad \mathbf{H}_2 = \frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^H}. \tag{B.19}$$

Matrix case

The extension from the vector gradient to matrix gradient is straightforward. For a real-differentiable $g(\mathbf{W}, \mathbf{W}^*) : \mathbb{C}^{N \times N} \times \mathbb{C}^{N \times N} \to \mathbb{R}$, we can write the first-order expansion as

$$\begin{aligned} \Delta g &= \left\langle \Delta \mathbf{W}, \tfrac{\partial g}{\partial \mathbf{W}^*} \right\rangle + \left\langle \Delta \mathbf{W}^*, \tfrac{\partial g}{\partial \mathbf{W}} \right\rangle = \\ &= 2\mathrm{Re}\left\{ \left\langle \Delta \mathbf{W}, \tfrac{\partial g}{\partial \mathbf{W}^*} \right\rangle \right\} \end{aligned} \tag{B.20}$$

where $\partial g/\partial \mathbf{W}$ is an $N \times N$ matrix whose (i,j)-th entry is the partial derivative of g with respect to w_{ij}. By arranging the matrix gradient into a vector and by using the Cauchy- Schwarz-Bunyakovski inequality [128], it is easy to show that the matrix gradient $\partial g/\partial \mathbf{W}^*$ defines the direction of the maximum rate of change in g with respect to $\mathbf{W}$.

For local stability analysis, Taylor expansions up to the second order is also frequently needed. Since the first-order matrix gradient takes a matrix form already, here we only provide the second-order expansion with respect to every entry of matrix $\mathbf{W}$. From (B.13), we obtain

$$\begin{array}{c} \Delta^2 g = \frac{1}{2}\left(\sum \frac{\partial g}{\partial w_{ij}\partial w_{kl}} dw_{ij} dw_{kl} + \sum \frac{\partial g}{\partial w_{ij}^* \partial w_{kl}^*} dw_{ij}^* dw_{kl}^*\right) + \\ + \sum \frac{\partial g}{\partial w_{ij}\partial w_{kl}^*} dw_{ij} dw_{kl}^* \end{array} \tag{B.21}$$

We can use the first-order Taylor series expansion to derive the relative gradient update rule for the complex case, which is usually directly extended to the complex case without a derivation. To write the relative gradient rule, we consider an update of the parameter matrix $\mathbf{W}$ in the invariant form $(\mathbf{\Delta W})\mathbf{W}$. We then write the first-order Taylor series expansion for the perturbation $(\mathbf{\Delta W})\mathbf{W}$ as

$$\begin{array}{c} \Delta g = \left\langle (\Delta \mathbf{W})\,\mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \right\rangle + \left\langle (\Delta \mathbf{W}^*)\,\mathbf{W}^*, \frac{\partial g}{\partial \mathbf{W}} \right\rangle = \\ = 2\mathrm{Re}\left\{ \left\langle \Delta \mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \mathbf{W}^H \right\rangle \right\} \end{array} \tag{B.22}$$

to determine the quantity that maximizes the rate of change in the function. The complex relative gradient of g at $\mathbf{W}$ is then written as $(\partial g/\partial \mathbf{W}^*)\mathbf{W}^H$ to write the relative gradient update term as

$$\Delta \mathbf{W} = -\mu \frac{\partial g}{\partial \mathbf{W}^*} \mathbf{W}^H \mathbf{W} \tag{B.23}$$

Upon substitution of $\mathbf{\Delta W}$ into (B.20), we observe that $\Delta g = -2\mu \left\| (\partial g/\partial \mathbf{W}^*)\mathbf{W}^H \right\|^2_{Fro}$ is a nonpositive quantity, thus a proper update term. The relative gradient can be regarded as a special case of natural gradient [6] in the matrix space, but provides the additional advantage that it can be easily extended to nonsquare matrices. It is possible to show how the relative gradient update rule for independent component analysis based on maximum likelihood can be derived in a very straightforward manner in the complex domain using (B.23) and Wirtinger calculus.

B.4 Complex random variables

A n-variate complex random vector (r.vc.) $\mathbf{x}$ is defined as an r.vc. of the form

$$\mathbf{x} = \mathbf{x}_R + j\mathbf{x}_I \tag{B.24}$$

where $\mathbf{x}_R$ and $\mathbf{x}_I$ are n-variate real r.vc.s, i.e., measurable functions from a probability space to $\mathbb{R}^n$. This is equivalent for $\mathbf{x}$ to be measurable from the probability space into $\mathbb{C}^n$ due to the separability of the complex space. Therefore, the probabilistic structure of the r.vc.s in $\mathbb{C}^n$ and the probabilistic structure of the r.vc.s in $\mathbb{R}^{2n}$ is the same. However, the *operator structure is different* as it is evident from the previous section. This gives distinct properties to the r.vc.s with complex values, and justifies studying them separately. Throughout this appendix all complex r.vc.s are assumed to be *full*. This means that the support of the induced measure of a n-dimensional r.vc. is not contained in any lower dimensional complex subspace.

Since the probabilistic structures of r.vc.s in $\mathbb{C}^n$ and in $\mathbb{R}^{2n}$ are the same, also the operator structure of r.vc.s in $\mathbb{C}^n$ can be studied by first using the isomorphism (B.4) and then applying the concepts associated with the real r.vc.s. However, we define these associated concepts directly on $\mathbb{C}^n$, since this approach is notationally more convenient. The *expectation* $E[\cdot]$ of a complex r.vc. $\mathbf{x}$ is defined as

$$E_{\mathbf{x}}[\mathbf{x}] = E_{\mathbf{x}_R}[\mathbf{x}_R] + jE_{\mathbf{x}_I}[\mathbf{x}_I] \tag{B.25}$$

and the *distribution function* or *cumulative density function* (*cdf*) $F_{\mathbf{x}}$ is given as $F_{\mathbf{x}}(\mathbf{z}) \stackrel{\Delta}{=} F_{\mathbf{x}_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}})$, where $\mathbf{z} = (z_1, \ldots, z_n)^T \in \mathbb{C}^n$ and $F_{\mathbf{x}_{\mathbb{R}}}$ denotes the distribution of real-valued r.vc. $\mathbf{x}_{\mathbb{R}}$. Then for independent r.v.s $\mathbf{s} = (s_1, \ldots, s_n)^T$, we have

$$F_s(z) = F_{s_{\mathbb{R}}}(z_{\mathbb{R}}) = \prod_{k=1}^{n} F_{(s_k)_{\mathbb{R}}}((z_k)_{\mathbb{R}}) = \prod_{k=1}^{n} F_{s_k}(z_k) \tag{B.26}$$

The same way we define the *probability density function* or *pdf* $f_{\mathbf{x}}$ (if it exists) of a n-dimensional complex r.vc. as $f_{\mathbf{x}}(\mathbf{z}) \stackrel{\Delta}{=} f_{\mathbf{x}_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}})$, and the *characteristic function* (*cf*) [116] as

$$\begin{aligned} \varphi_x(\mathbf{z}) \stackrel{\Delta}{=} \varphi_{x_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}}) = E_{x_{\mathbb{R}}}\left[\exp\left(j\left\langle \mathbf{z}_{\mathbb{R}}, x_{\mathbb{R}}\right\rangle\right)\right] = \\ = E_x\left[\exp\left(j\mathrm{Re}\left\{\mathbf{z}, x\right\}\right)\right] \end{aligned} \tag{B.27}$$

It follows directly from (B.26) that for independent complex r.v.s $\mathbf{s} = (s_1, \ldots, s_n)^T$

$$\varphi_{\mathbf{s}}(\mathbf{z}) = \prod_{k=1}^{n} \varphi_{s_k}(z_k) \tag{B.28}$$

Using a standard property of real cf.s and the properties of the isomorphism (B.4), we have a useful relation for the cf. of an r.vc. $\mathbf{x}$ and the cf. of the linearly transformed r.vc. $\mathbf{Cx}$. Namely, for any complex matrix $\mathbf{C}$, we have

$$\begin{aligned} \varphi_{\mathbf{Cx}}(\mathbf{z}) = \varphi_{(\mathbf{Cx})_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}}) = \varphi_{\mathbf{C}_{\mathbb{R}}\mathbf{x}_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}}) = \varphi_{\mathbf{x}_{\mathbb{R}}}\left(\left(\mathbf{C}_{\mathbb{R}}\right)^T \mathbf{z}_{\mathbb{R}}\right) = \\ = \varphi_{\mathbf{x}_{\mathbb{R}}}\left(\left(\mathbf{C}^H\right)_{\mathbb{R}} \mathbf{z}_{\mathbb{R}}\right) = \varphi_{\mathbf{x}_{\mathbb{R}}}\left(\left(\mathbf{C}^H\mathbf{z}\right)_{\mathbb{R}}\right) = \varphi_{\mathbf{x}}\left(\mathbf{C}^H\mathbf{z}\right) \end{aligned} \tag{B.29}$$

Finally, a cf. $\varphi_{\mathbf{x}}(\mathbf{z})$ is called *analytic* if $\varphi_{\mathbf{x}_{\mathbb{R}}}(\mathbf{z}_{\mathbb{R}})$ is an analytic cf., i.e., the real cf. has a regular extension defined on in some neighborhood of the origin.

B.4.1 Second-Order Statistics of Complex Random Vectors

An r.vc. $\mathbf{x}$ has *finite second order* or *weak second order* statistics if $E_{\mathbf{x}}\left[|\langle \mathbf{x}, \mathbf{z}\rangle|^2\right] < \infty$ for all $\mathbf{z} \in \mathbb{C}^n$. This is clearly equivalent to the existence of finite second order statistics for both real r.vc.s $\mathbf{x}_R$ and $\mathbf{x}_I$. All r.vc.s in this section are assumed to have finite second order statistics. Such r.vc.s are in general called *second-order* complex r.vc.s.

The second-order statistics between two real r.vc.s may be described by the covariance matrix. The complex *covariance matrix* $\mathrm{cov}[\mathbf{x}_1, \mathbf{x}_2]$ of two complex r.vc.s $\mathbf{x}_1$ and $\mathbf{x}_2$ may be defined as

$$\mathrm{cov}\left[\mathbf{x}_1, \mathbf{x}_2\right] \triangleq E_{\mathbf{x}_1,\mathbf{x}_2}\left[\left(\mathbf{x}_1 - \mathbf{m}_{\mathbf{x}_1}\right)\left(\mathbf{x}_2 - \mathbf{m}_{\mathbf{x}_2}\right)^H\right] \tag{B.30}$$

where $\mathbf{m}_{\mathbf{x}1} = E_{\mathbf{x}_1}[\mathbf{x}_1]$ and $\mathbf{m}_{\mathbf{x}2} = E_{\mathbf{x}_2}[\mathbf{x}_2]$.

However, considering the real representations of the complex r.vc.s, it can be seen that the complex covariance matrix does not give complete second order description. For that we define the *pseudo-covariance matrix* $\mathrm{pcov}[\mathbf{x}_1, \mathbf{x}_2]$ [148] as

$$\mathrm{pcov}\left[\mathbf{x}_1, \mathbf{x}_2\right] \triangleq E_{\mathbf{x}_1,\mathbf{x}_2}\left[\left(\mathbf{x}_1 - \mathbf{m}_{\mathbf{x}_1}\right)\left(\mathbf{x}_2 - \mathbf{m}_{\mathbf{x}_2}\right)^T\right] = \mathrm{cov}\left[\mathbf{x}_1, \mathbf{x}_2^*\right] \tag{B.31}$$

Two complex r.vc.s $\mathbf{x}_1$ and $\mathbf{x}_2$ are *uncorrelated* if real r.vc.s $(\mathbf{x}_1)_{\mathbb{R}}$ and $(\mathbf{x}_2)_{\mathbb{R}}$ are uncorrelated, i.e.,

$$\mathrm{cov}\left[(\mathbf{x}_1)_{\mathbb{R}}, (\mathbf{x}_2)_{\mathbb{R}}\right] = \mathbf{O}_{2n\times 2n} \tag{B.32}$$

where $\mathbf{O}_{2n\times 2n}$ denotes the $2n \times 2n$ matrix of zeros. Then, by using the properties from the previous section, the following lemma follows directly.

Lemma 13 *Complex r.vc.s $\mathbf{x}_1$ and $\mathbf{x}_2$ are uncorrelated if and only if* $\mathrm{cov}[\mathbf{x}_1, \mathbf{x}_2] = \mathrm{pcov}[\mathbf{x}_1, \mathbf{x}_2] = \mathbf{O}_{\mathrm{n}\times\mathrm{n}}$.

As it is the case with real r.vc.s, the internal correlation structure of a single r.vc. $\mathbf{x}$ may be of interest in addition to correlation between two r.vc.s. Then we define $\mathrm{cov}[\mathbf{x}] \triangleq \mathrm{cov}[\mathbf{x}, \mathbf{x}]$ and $\mathrm{pcov}[\mathbf{x}] \triangleq \mathrm{pcov}[\mathbf{x}, \mathbf{x}]$, and call them the covariance matrix and the pseudo-covariance matrix of an r.vc. $\mathbf{x}$, respectively. It is easily seen that the covariance matrix $\mathrm{cov}[\mathbf{x}]$ is Hermitian and the pseudo-covariance matrix $\mathrm{pcov}[\mathbf{x}]$ is symmetric. Since all r.vc.s are assumed to be full, the covariance matrix $\mathrm{cov}[\mathbf{x}]$ is also positive definite. R.vc. $\mathbf{x}$ is said to have *uncorrelated components* if all its marginal r.v.s x_k and x_l, $k \neq l$, are uncorrelated. The following lemma is a simple consequence of Lemma 13.

Lemma 14 *A complex r.vc. $\mathbf{x}$ has uncorrelated components if and only if its covariance matrix and pseudocovariance matrix are diagonal.*

An r.vc. $\mathbf{x}$ is said to be *spatially white*, if $\mathrm{cov}[\mathbf{x}] = \sigma^2\mathbf{I}_\mathrm{n}$ for some $\sigma > 0$. If $\mathrm{pcov}[\mathbf{x}] = \mathbf{O}_{\mathrm{n}\times\mathrm{n}}$, then the r.vc. is called *second-order circular* (or *circularly symmetric*). Some authors prefer the term *proper* [148, 203, 134]. A stronger definition of *circularity* is based on the pdf of the complex random variable such that for any α, the pdfs of the r.v. x and $e^{j\alpha}x$ are the same. Circular r.vc.s have gained most of the attention in the literature of complex r.vc.s. This is likely due to the fact that all the second order information of circular r.vc.s is contained in the covariance matrix, which, on the other hand, behaves like the covariance matrix for the real r.vc.s. However, in this appendix we need the complete second order description to be derived next. Our approach is to our best knowledge novel, mainly based on the following theorem.

Theorem 26 *Any full complex n-dimensional r.vc. $\mathbf{x}$ with finite second order statistics can be transformed by using a nonsingular square matrix $\mathbf{C}$ such that the r.vc. $\mathbf{s} = (s_1, \ldots, s_n)^T = \mathbf{C}\mathbf{x}$ has the following properties [203]:*

1. $\mathrm{cov}[\mathbf{s}] = \mathbf{I}_\mathrm{n}$;
2. $\mathrm{pcov}[\mathbf{s}] = \mathrm{diag}(\lambda[\mathbf{s}])$, *where* $\lambda[\mathbf{s}] = (\lambda_1, \ldots, \lambda_n)^T$ *denotes a vector such that* $\lambda_1 \geq \ldots \geq \lambda_n$.

Since $\mathrm{cov}[\mathbf{x}] = \mathrm{cov}[\mathbf{x}_\mathrm{R}]+\mathrm{cov}[\mathbf{x}_\mathrm{I}]$ and $\mathrm{pcov}[\mathbf{x}] = \mathrm{cov}[\mathbf{x}_\mathrm{R}]-\mathrm{cov}[\mathbf{x}_\mathrm{I}]+2\mathrm{jcov}[\mathbf{x}_\mathrm{R}, \mathbf{x}_\mathrm{I}]$ for any r.v. $x = x_R+jx_I$, it follows by Theorem 26 that $\mathrm{cov}[\mathrm{Re}\{\mathrm{s_k}\}, \mathrm{Im}\{\mathrm{s_k}\}] = 0$ and $1 \geq \lambda_k = \mathrm{cov}[\mathrm{Re}\{\mathrm{s_k}\}] - \mathrm{cov}[\mathrm{Im}\{\mathrm{s_k}\}] \geq 0$, $k = 1, \ldots, n$.

The r.vc.s satisfying the properties of Theorem 26 have a special structure, and they are here called *strongly uncorrelated*. Any strongly uncorrelated r.vc. is white with $\mathrm{cov}[\mathbf{s}] = \mathbf{I}_\mathrm{n}$, but the converse is not true. In general, for a given r.vc. $\mathbf{x}$, the strongly uncorrelated r.vc. $\mathbf{s}$ and the *strong-uncorrelating transform* $\mathbf{C}$ given by Theorem 26 are not unique. However, we have the following.

Theorem 27 *For a given r.vc. $\mathbf{x}$, the vector $\lambda[\mathbf{s}]$ in Theorem 26 is unique.*

The previous theorems lead to a useful characterization of second-order complex r.vc.s.

Definition 39 *The vector $\lambda[\mathbf{x}] \triangleq \lambda[\mathbf{s}] = (\lambda_1, \ldots, \lambda_n)^T$ in Theorem 26 is called the circularity spectrum of an r.vc. $\mathbf{x}$. An element of the circularity spectrum corresponding to an r.v. is called a circularity coefficient.*

Any r.vc. $\mathbf{x}$ is clearly second order circular if and only if its circularity spectrum is a zero vector, i.e., $\lambda[\mathbf{x}] = \mathbf{O}_{n\times 1}$.

Corollary 3 *If the circularity spectrum of an r.vc. has distinct elements, all rows corresponding to nonzero circularity coefficients of the strong-uncorrelating transform are unique up to multiplication of the row by -1. A row corresponding to the zero coefficient is unique up to multiplication of the row by $e^{j\theta}$, $\theta \in \mathbb{R}$.*

Lemma 15 *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be uncorrelated second-order complex r.v.s. Then*

1. $$0 \leq \lambda\left[c\mathbf{x}\right] = \lambda\left[\mathbf{x}\right] = \frac{\left|\mathrm{pcov}\left[\mathbf{x}\right]\right|}{\mathrm{cov}\left[\mathbf{x}\right]} \leq 1$$
   for any nonzero constant $c \in \mathbb{C}$*;*

2. $\lambda[\mathbf{x}] = 1$ *if and only if* $x = c(s_R + j\alpha)$ *for some unit variance real r.v.* s_R *and deterministic constants* $0 \neq c \in \mathbb{C}$, $\alpha \in \mathbb{R}$*;*

3. $$\lambda\left[\mathbf{x}+\mathbf{y}\right] = \frac{\left|\mathrm{pcov}\left[\mathbf{x}\right] + \mathrm{pcov}\left[\mathbf{y}\right]\right|}{\mathrm{cov}\left[\mathbf{x}\right] + \mathrm{cov}[\mathbf{y}]} \leq \max\left\{\lambda\left[\mathbf{x}\right], \lambda\left[\mathbf{y}\right]\right\}$$
   with the equality if and only if $\lambda[\mathbf{x}] = \lambda[\mathbf{y}]$ *and* $\mathrm{Arg}(\mathrm{pcov}[\mathrm{x}]) = \mathrm{Arg}(\mathrm{pcov}[\mathrm{y}])$ *if* $\lambda[\mathbf{x}] \neq 0$*.*

B.4.2 Complex Normal Random Vectors

There are no commonly agreed definitions of what is meant by complex normal r.vc.s. It is natural to require that a r.vc. $\mathbf{x}$ is *normal* (*Gaussian*) if the real r.vc. $\mathbf{x}_{\mathbb{R}}$ is multivariate normal. Such r.vc.s are generally called *wide sense normal* r.vc.s [203]. Since the real complex normal r.vc. is completely characterized by its mean vector and covariance, the results from the previous section show that a wide sense complex normal r.vc. is completely specified by its mean, covariance matrix, and pseudo-covariance matrix.

However, all wide sense normal r.vc.s do not possess all the properties that real normal r.vc.s do. Only a special subclass of wide sense normal r.vc.s has a density function similar to the real r.vc.s, i.e. maximizes the entropy. Such r.vc.s are called *narrow sense normal* r.vc.s [203]. They are wide sense normal r.vc.s such that the real and imaginary parts of any linear projection of the r.vc. are independent and have equal variances. This condition is equivalent to the requirement that a wide sense normal r.vc. is second order circular (see, e.g., [134]).

In order to establish the properties of the complex ICA model, neither wide sense normal in its full generality nor narrow sense normal is adequate, and a more specific characterization of complex normal r.vc.s is needed.

The main result is the following decomposition theorem for complex normal random vectors.

Theorem 28 *A r.vc.* $\mathbf{n}$ *is complex normal with circularity spectrum* λ *if and only if*

$$\mathbf{n} = \mathbf{C}\left(\eta_R + j\eta_I\right) + \mu \tag{B.33}$$

for some nonsingular matrix $\mathbf{C}$*, a complex constant vector* μ*, and multinormal real independent r.vc.s* $\eta_R \sim N\left(\mathbf{0}_{n\times 1}, \frac{1}{2}\mathbf{I}_n + \frac{1}{2}\mathrm{diag}\left(\lambda\right)\right)$ *and* $\eta_I \sim N\left(\mathbf{0}_{n\times 1}, \frac{1}{2}\mathbf{I}_n - \frac{1}{2}\mathrm{diag}\left(\lambda\right)\right)$*. Also* $\mathrm{cov}[\mathbf{n}] = \mathbf{C}\mathbf{C}^{\mathrm{H}}$, $\mathrm{pcov}[\mathbf{n}] = \mathbf{C}\mathrm{diag}(\lambda)\mathbf{C}^{\mathrm{T}}$*, and* $E_{\mathbf{n}}[\mathbf{n}] = \mu$*.*

A complex normal r.vc. η such that $\mathbf{C} = \mathbf{I}_n$ and $\mu = \mathbf{0}_{n\times 1}$ in the representation (B.33), i.e., $\eta = \eta_R + j\eta_I$, is called *standard complex normal* with the circularity spectrum λ. Clearly any centered and strongly uncorrelated complex normal r.vc. is standard. Also, it is seen that any complex normal r.vc. may be alternatively specified by the mean, the circularity spectrum, and the (inverse of) strong-uncorrelating matrix $\mathbf{C}$.

The previous decomposition allows the derivation of *differential entropy* of a complex normal r.vc. in a closed form. Entropy $h(\mathbf{n})$ of an r.vc. $\mathbf{x}$ is defined as the entropy [55] of the real r.vc. $\mathbf{x}_{\mathbb{R}}$. The following result has been implicitly derived without reference to circularity coefficients.

Corollary 4 *The differential entropy $h(\mathbf{n})$ of a zero-mean complex normal r.vc. $\mathbf{n}$ with the circularity coefficients $\lambda_k \neq 1$, $k = 1, \ldots, n$ is given by*

$$h(\mathbf{n}) = \log\left(\det\left(\pi e \mathrm{cov}\left[\mathbf{n}\right]\right)\right) + \frac{1}{2}\sum_{k=1}^{n}\log\left(1-\lambda_k^2\right) \tag{B.34}$$

Since the summation term on the right of (B.34) is always nonpositive and the entropy of real r.vc.s with the given covariance is maximized for Gaussian r.vc.s [55], it may be seen that the entropy of complex r.vc.s with the given covariance is maximized for a narrow sense complex normal r.vc. [134], i.e., for a complex normal r.vc. with zero pseudo-covariance. Theorem 28 allows also an easy derivation of the cf. of a complex normal r.vc. [149], [203].

Corollary 5 *The characteristic function of a complex normal r.vc. $\mathbf{n}$ is given by*

$$\begin{aligned}\varphi_{\mathbf{n}}(\mathbf{z}) &= \exp\left(-\tfrac{1}{4}\mathbf{z}^H \mathrm{cov}\left[\mathbf{n}\right]\mathbf{z} - \tfrac{1}{4}\mathrm{Re}\left\{\mathbf{z}^H \mathrm{pcov}\left[\mathbf{n}\right]\mathbf{z}^*\right\} + j\mathrm{Re}\left\{\mathbf{z}^H E_{\mathbf{n}}\left[\mathbf{n}\right]\right\}\right) = \\ &= \exp\left(-\tfrac{1}{4}\left(\mathrm{Re}\left\{\left\langle \mathbf{z}, \mathrm{cov}\left[\mathbf{n}\right]\mathbf{z} + \mathrm{pcov}\left[\mathbf{n}\right]\mathbf{z}^*\right\rangle\right\} + j\mathrm{Re}\left\{\left\langle \mathbf{z}, E_{\mathbf{n}}\left[\mathbf{n}\right]\right\rangle\right\}\right)\right)\end{aligned} \tag{B.35}$$

Corollary 5 shows in particular that the *second characteristic function* $\psi_{\mathbf{x}} \triangleq \log\varphi_{\mathbf{x}}$ of a complex r.vc. $\mathbf{x}$ is a second-order wide sense polynomial in variables $(\mathbf{z}, \mathbf{z}^*)$. Theorem 28 can be also used to derive the density function of a complex normal r.vc. However, unlike the cf., the density function of a wide sense normal r.vc. does not appear to have a simple form. See [149] for expressions for the density function in terms of the covariance and the pseudo-covariance matrices.

Example 11.

Let the components of $\mathbf{n}$ be uncorrelated complex normal r.v.s with the same circularity coefficient λ. Now for a diagonal matrix $\mathbf{\Lambda}$ the r.vc. $\mathbf{\Lambda n}$ is standard complex normal with the circularity spectrum $(\lambda, \ldots, \lambda)^T$, and for any (realvalued) orthonormal matrix $\mathbf{O}$

$$\mathrm{cov}\left[\mathbf{O\Lambda n}\right] = \mathbf{O}\mathrm{cov}\left[\mathbf{\Lambda n}\right]\mathbf{O}^H = \mathbf{O}\mathbf{I}_n\mathbf{O}^T = \mathbf{I}_n$$

and

$$\mathrm{pcov}\left[\mathbf{O\Lambda n}\right] = \mathbf{O}\mathrm{pcov}\left[\mathbf{\Lambda n}\right]\mathbf{O}^T = \mathbf{O}\left(\lambda\mathbf{I}_n\right)\mathbf{O}^T = \lambda\mathbf{I}_n$$

Therefore, the r.vc. $\mathbf{O\Lambda n}$ is also standard complex normal.

B.5 Some useful theorems for complex random variables

The following theorem is a direct consequence of the multivariate version of the real Marcinkiewicz theorem. The theorem shows essentially that a complex normal r.v. is the only r.v. whose second cf. is a wide sense polynomial.

Theorem 29 (Complex Marcinkiewicz) *If in some neighborhood of zero the cf. $\varphi_{\mathbf{x}}$ of a complex r.v. $\mathbf{x}$ admits the representation*

$$\varphi_{\mathbf{x}}\left(z\right) = \exp\left(\mathcal{P}\left(z, z^*\right)\right) \tag{B.36}$$

where $\mathcal{P}$ is a wide sense polynomial, then the r.v. $\mathbf{x}$ is complex normal.

Also the well-known Cramer's theorem has a direct complex counterpart.

Theorem 30 (Complex Cramer) *If $\mathbf{s}_1$ and $\mathbf{s}_2$ are independent r.v.s such that $\mathbf{s_1} + \mathbf{s_2}$ is a complex normal r.v., then each of the r.v.s $\mathbf{s}_1$ and $\mathbf{s}_2$ is complex normal.*

Lemma 16 *Consider the equation, assumed valid for $|z_1|, |z_2| < \epsilon$*

$$\sum_{k=1}^{n} \psi_k\left(z_1 + c_k z_2\right) = h_1\left(z_1\right) + h_2\left(z_2\right) \tag{B.37}$$

where ψ_k, $k = 1, \ldots, n$, h_1 and h_2 are continuous complex-valued functions of complex variables and the nonzero complex numbers c_k, $k = 1, \ldots, n$, are distinct. Then all the functions in (B.37) *are wide sense polynomials in (z, z^*) of degree not exceeding.*

One of the main characterization theorems for real r.v.s is the well-known *Darmois-Skitovich* theorem. The theorem is fundamental for proving the identifiability of real ICA models [50], [64]. Here we extend the theorem to complex r.v.s.

The complex extension of Darmois-Skitovich theorem has exactly the same form as the real theorem with the wide sense complex normal r.v.s taking the role of real normal r.v.s. Hence, this theorem is an example where the analogy between theories of narrow sense complex normal r.v.s and real normal r.v.s is broken.

Theorem 31 (Complex Darmois-Skitovich) *Let $s_1, \ldots, n$ be mutually independent complex r.v.s. If the linear forms (the r.v.s)*

$$x_1 = \sum_{k=1}^{n} \alpha_k s_k$$

and

$$x_2 = \sum_{k=1}^{n} \beta_k s_k$$

where $\alpha_k, \beta_k \in \mathbb{C}$, $k = 1, \ldots, n$ are independent, then r.v.s s_k for which $\alpha_k \beta_k \neq 0$ are complex normal.

Although narrow sense complex normal r.v.s had to be admitted to the complex Darmois-Skitovich theorem, it may still appear in the view of Corollary 3 that complex normal r.v.s appearing in the theorem can not be completely arbitrary. That is, it may appear that some of the circularity coefficients of normal r.v.s should be equal. It is true if $n = 2$. However, it is not generally true as it is shown in the next example.

Example 12.

Let $\eta_1 = (n_1, n_2, n_3)^T$ be standard complex normal r.vc. with the circularity spectrum $\lambda[\eta_1]$. Then

$$\eta_2 = \frac{1}{5\sqrt{2}} \begin{pmatrix} 3 & 5 & 4 \\ 3 & -5 & 4 \end{pmatrix} \eta_1$$

is also standard complex normal r.vc. with the circularity spectrum $\lambda[\eta_2] = \begin{pmatrix} \frac{1}{5} & \frac{1}{5} \end{pmatrix}^T$. Thus marginals of η_2 are independent, and the Darmois-Skitovich theorem applies. However, the circularity spectrum of η_1 is distinct. Notice also that by Example 12, the r.vc. obtained from η_2 by multiplying with any orthogonal matrix is also standard complex normal r.vc. with the same circularity spectrum.

Other useful results are the following.

Lemma 17 *Suppose independent complex r.v.s s_1 and s_2 are independent of complex normal r.v.s n_1 and n_2. If $s_1 + n_1$ is independent of $s_2 + n_2$, then also n_1 and n_2 are independent.*

Lemma 18 *If complex r.v.s n and s are independent and $n + s$ is independent of n, then n is degenerate (i.e., a constant).*

C Proofs

—The truth is rarely pure, and never simple.
Oscar Wilde

In the following rows I will present a mathematical derivation of the learning rules for the weights matrix $\mathbf{W}$ and for the control points adaptation.
First it is explained the feed-forward phase and the then backward phase, so I can derive the w-rule for the weights matrix. Finally it is derived the control points adaptation. The algorithm is derived by maximizing the mutual entropy of the output vector $\mathbf{y}$.

Then the Amari's natural gradient adaptation rule is rapidly presented.

C.1 The Linear Case

C.1.1 The feed-forward phase

Here we present the proofs of (7.17) and (7.19). For notation we use the symbols in Figure C.1. See also [165].

All the signals in this architecture are complex-valued signals, and the i-th observed signal is represented by: $x_i = x_{iR} + jx_{iI}$. The complex unmixing matrix is the following:

$$\mathbf{W} = \mathbf{W}_R + j\mathbf{W}_I = \\ = \begin{pmatrix} w_{11R} + jw_{11I} & \cdots & w_{1jR} + jw_{1jI} & \cdots & w_{1NR} + jw_{1NI} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i1R} + jw_{i1I} & \cdots & w_{ijR} + jw_{ijI} & \cdots & w_{iNR} + jw_{iNI} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{N1R} + jw_{N1I} & \cdots & w_{NjR} + jw_{NjI} & \cdots & w_{NNR} + jw_{NNI} \end{pmatrix}$$

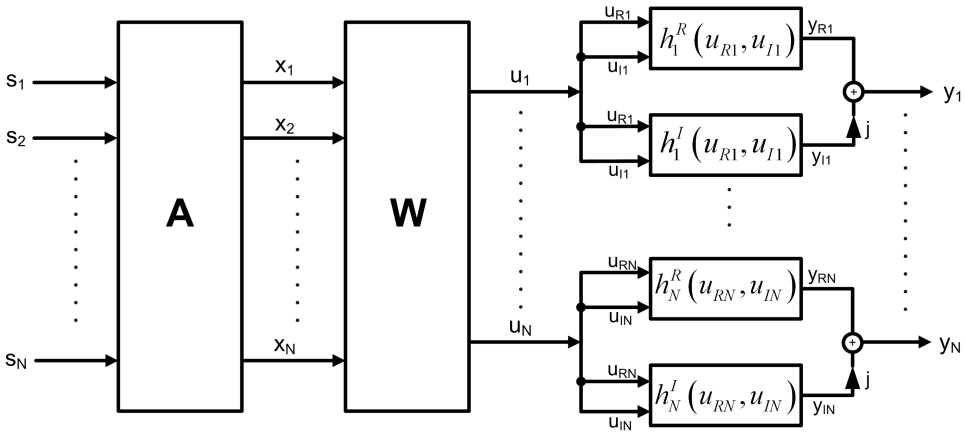

Fig. C.1: The entire system

The output of the un-mixing matrix $\mathbf{W}$, the signal u_i, is a linear combination of the observed signals and the un-mixing matrix elements:

$$\begin{aligned} u_i &= u_{iR} + ju_{iI} = \sum_{k=1}^{N} \left(w_{ijR} + jw_{ijI}\right)\left(x_{jR} + jx_{jI}\right) = \\ &= \sum_{k=1}^{N} \left(w_{ijR}x_{jR} - w_{ijI}x_{jI} + j\left(w_{ijR}x_{jI} + w_{ijI}x_{jR}\right)\right) = \\ &= \sum_{k=1}^{N} \left(w_{ijR}x_{jR} - w_{ijI}x_{jI}\right) + j\sum_{k=1}^{N} \left(w_{ijR}x_{jI} + w_{ijI}x_{jR}\right) \end{aligned} \tag{C.1}$$

Then the signal (C.1) achieves the complex spline function (the Catmull-Rom spline in this case) and we obtain the output signal y_i described by:

$$y_i = y_{iR} + jy_{iI} = u_i\left(u_{iR}, u_{iI}\right) + jv_i\left(u_{iR}, u_{iI}\right) \tag{C.2}$$

C.1.2 The backward phase

For the Infomax principle, proposed by Bell & Sejnowski [15], we can reach the separation of the input complex-valued signals maximizing the joint entropy [55] of the output signal vector

$$\mathbf{y} = \mathbf{y}_R + j\mathbf{y}_I \tag{C.3}$$

composed by the N stochastic complex variables (C.2).

This vector has a probability density function (pdf) given by

$$p_{\mathbf{y}}\left(y_{1R}, \ldots, y_{NR}, y_{1I}, \ldots, y_{NI}\right) = \prod_{i=1}^{N} p_{y_i}\left(y_{iR}, y_{iI}\right) \tag{C.4}$$

if the N stochastic variable are independent.

Using expression (C.4) the joint entropy of the output vector (C.3) is [144]:

$$\begin{aligned} &H(\mathbf{y}) = -\int p_{\mathbf{y}}(y_{1R},\ldots,y_{NR},y_{1I},\ldots,y_{NI})\cdot \\ &\cdot \ln\left(p_{\mathbf{y}}(y_{1R},\ldots,y_{NR},y_{1I},\ldots,y_{NI})\right)\cdot dy_{1R},\ldots,dy_{NR},dy_{1I},\ldots,dy_{NI} = \\ &\overset{def}{=} -E\left(\ln\left(p_{\mathbf{y}}(y_{1R},\ldots,y_{NR},y_{1I},\ldots,y_{NI})\right)\right) = \\ &= E\left(-\ln\left(p_{\mathbf{y}}(y_{1R},\ldots,y_{NR},y_{1I},\ldots,y_{NI})\right)\right) \end{aligned} \tag{C.5}$$

For (C.5) the Infomax principle is simply:

$$\max_{\mathbf{W}} E\left(-\ln\left(p_{\mathbf{y}}(y_{1R},\ldots,y_{NR},y_{1I},\ldots,y_{NI})\right)\right) \tag{C.6}$$

Because the law between $\mathbf{y}$ (the output vector) and $\mathbf{x}$ (the input vector) is the un-mixing matrix $\mathbf{W}$ and $\mathbf{u}$, we can derive the probability density function of the output vector from the probability density function of the input vector by [144]:

$$p\left(\begin{bmatrix}\mathbf{y}_R\\ \mathbf{y}_I\end{bmatrix}\right) = \frac{p\left(\begin{bmatrix}\mathbf{x}_R\\ \mathbf{x}_I\end{bmatrix}\right)}{\det(\mathbf{J})} \tag{C.7}$$

where $\mathbf{J}$ is the Jacobian of the transformation:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial y_{1R}}{\partial x_{1R}} & \cdots & \frac{\partial y_{1R}}{\partial x_{NR}} & \frac{\partial y_{1R}}{\partial x_{1I}} & \cdots & \frac{\partial y_{1R}}{\partial x_{NI}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{NR}}{\partial x_{1R}} & \cdots & \frac{\partial y_{NR}}{\partial x_{NR}} & \frac{\partial y_{NR}}{\partial x_{1I}} & \cdots & \frac{\partial y_{NR}}{\partial x_{NI}} \\ \frac{\partial y_{1I}}{\partial x_{1R}} & \cdots & \frac{\partial y_{1I}}{\partial x_{NR}} & \frac{\partial y_{1I}}{\partial x_{1I}} & \cdots & \frac{\partial y_{1I}}{\partial x_{NI}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{NI}}{\partial x_{1R}} & \cdots & \frac{\partial y_{NI}}{\partial x_{NR}} & \frac{\partial y_{NI}}{\partial x_{1I}} & \cdots & \frac{\partial y_{NI}}{\partial x_{NI}} \end{pmatrix} \tag{C.8}$$

The single elements of (C.8) are calculated using (C.1) as:

$$\begin{aligned} \frac{\partial y_{iR}}{\partial x_{jR}} &= \frac{\partial y_{iR}}{\partial u_{iR}}\frac{\partial u_{iR}}{\partial x_{jR}} + \frac{\partial y_{iR}}{\partial u_{iI}}\frac{\partial u_{iI}}{\partial x_{jR}} = \frac{\partial y_{iR}}{\partial u_{iR}}w_{ijR} + \frac{\partial y_{iR}}{\partial u_{iI}}w_{ijI} \\ \frac{\partial y_{iR}}{\partial x_{jI}} &= \frac{\partial y_{iR}}{\partial u_{iR}}\frac{\partial u_{iR}}{\partial x_{jI}} + \frac{\partial y_{iR}}{\partial u_{iI}}\frac{\partial u_{iI}}{\partial x_{jI}} = \frac{\partial y_{iR}}{\partial u_{iR}}(-w_{ijI}) + \frac{\partial y_{iR}}{\partial u_{iI}}w_{ijR} \\ \frac{\partial y_{iI}}{\partial x_{jR}} &= \frac{\partial y_{iI}}{\partial u_{iR}}\frac{\partial u_{iR}}{\partial x_{jR}} + \frac{\partial y_{iI}}{\partial u_{iI}}\frac{\partial u_{iI}}{\partial x_{jR}} = \frac{\partial y_{iI}}{\partial u_{iR}}w_{ijR} + \frac{\partial y_{iI}}{\partial u_{iI}}w_{ijI} \\ \frac{\partial y_{iI}}{\partial x_{jI}} &= \frac{\partial y_{iI}}{\partial u_{iR}}\frac{\partial u_{iR}}{\partial x_{jI}} + \frac{\partial y_{iI}}{\partial u_{iI}}\frac{\partial u_{iI}}{\partial x_{jI}} = \frac{\partial y_{iI}}{\partial u_{iR}}(-w_{ijI}) + \frac{\partial y_{iI}}{\partial u_{iI}}w_{ijR} \end{aligned} \tag{C.9}$$

Substituting the expressions (C.9) in the Jacobian (C.8), we can calculate it splitting the Jacobian $\mathbf{J}$ in the product of two matrices. In fact, applying the Cauchy-Riemann conditions [122]:

$$\begin{cases} \frac{\partial y_{iR}}{\partial u_{iR}} = \frac{\partial y_{iI}}{\partial u_{iI}} \\ \frac{\partial y_{iR}}{\partial u_{iI}} = -\frac{\partial y_{iI}}{\partial u_{iR}} \end{cases} \tag{C.10}$$

we obtain the following relation

$$\det(\mathbf{J}) = \det\begin{pmatrix} \frac{\partial y_{1R}}{\partial u_{1R}} & \cdots & 0 & -\frac{\partial y_{1I}}{\partial u_{1R}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial y_{NR}}{\partial u_{NR}} & 0 & \cdots & -\frac{\partial y_{NI}}{\partial u_{NR}} \\ \frac{\partial y_{1I}}{\partial u_{1R}} & \cdots & 0 & \frac{\partial y_{1R}}{\partial u_{1R}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial y_{NI}}{\partial u_{NR}} & 0 & \cdots & \frac{\partial y_{NR}}{\partial u_{NR}} \end{pmatrix} \cdot \tag{C.11}$$
$$\cdot \det\begin{pmatrix} w_{11R} & \cdots & w_{1NR} & -w_{11I} & \cdots & -w_{1NI} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{N1R} & \cdots & w_{NNR} & -w_{N1I} & \cdots & -w_{NNI} \\ w_{11I} & \cdots & w_{1NI} & w_{11R} & \cdots & w_{1NR} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{N1I} & \cdots & w_{NNI} & w_{N1R} & \cdots & w_{NNR} \end{pmatrix}$$

where the last matrix in (C.11) can be expressed as

$$\det\begin{pmatrix} w_{11R} & \cdots & w_{1NR} & -w_{11I} & \cdots & -w_{1NI} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{N1R} & \cdots & w_{NNR} & -w_{N1I} & \cdots & -w_{NNI} \\ w_{11I} & \cdots & w_{1NI} & w_{11R} & \cdots & w_{1NR} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{N1I} & \cdots & w_{NNI} & w_{N1R} & \cdots & w_{NNR} \end{pmatrix} = \tag{C.12}$$
$$= \det\begin{pmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{pmatrix} = \det\left(\tilde{\mathbf{W}}\right)$$

and the matrix $\tilde{\mathbf{W}}$ is in the canonical form.

The first determinant in (C.11) can be calculated remembered that (see [73])

$$\det\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \det(D) \cdot \det(A - BD^{-1}C) \tag{C.13}$$

where $A \in \mathbb{R}^{s\times s}$, $D \in \mathbb{R}^{r\times r}$ are invertible matrices, $B \in \mathbb{R}^{s\times r}$, $C \in \mathbb{R}^{r\times s}$. Hence, because $\det(AB) = \det(A) \cdot \det(B)$, we derive from (C.13)

$$\det(D)\cdot\det(A-BD^{-1}C) = \det(DA-DBD^{-1}C) = \det(DA-BC) \tag{C.14}$$

the last identity is justified for the diagonal nature of all matrices, thus the matrix product is commutative. Now because $A = D$ and $C = -B$ and the matrices are all diagonal, so that the n-th power of a diagonal matrix is a matrix with only the diagonal elements nonzero and each equal to the n-th

power of each element, from (C.14) we obtain directly:

$$
\begin{aligned}
&\det\begin{pmatrix}
\frac{\partial y_{1R}}{\partial u_{1R}} & \cdots & 0 & -\frac{\partial y_{1I}}{\partial u_{1R}} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \frac{\partial y_{NR}}{\partial u_{NR}} & 0 & \cdots & -\frac{\partial y_{NI}}{\partial u_{NR}} \\
\frac{\partial y_{1I}}{\partial u_{1R}} & \cdots & 0 & \frac{\partial y_{1R}}{\partial u_{1R}} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & \frac{\partial y_{NI}}{\partial u_{NR}} & 0 & \cdots & \frac{\partial y_{NR}}{\partial u_{NR}}
\end{pmatrix} = \\
&= \det\left\{\begin{bmatrix}
\left(\frac{\partial y_{1R}}{\partial u_{1R}}\right)^2 & 0 & \cdots & 0 \\
0 & \left(\frac{\partial y_{2R}}{\partial u_{2R}}\right)^2 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \left(\frac{\partial y_{NR}}{\partial u_{NR}}\right)^2
\end{bmatrix} + \right. \\
&\left. + \begin{bmatrix}
\left(\frac{\partial y_{1I}}{\partial u_{1R}}\right)^2 & 0 & \cdots & 0 \\
0 & \left(\frac{\partial y_{2I}}{\partial u_{2R}}\right)^2 & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \left(\frac{\partial y_{NI}}{\partial u_{NR}}\right)^2
\end{bmatrix}\right\} = \\
&= \det\begin{bmatrix}
\left(\frac{\partial y_{1R}}{\partial u_{1R}}\right)^2 + \left(\frac{\partial y_{1I}}{\partial u_{1R}}\right)^2 & 0 & \cdots & 0 \\
0 & \left(\frac{\partial y_{2R}}{\partial u_{2R}}\right)^2 + \left(\frac{\partial y_{2I}}{\partial u_{2R}}\right)^2 & \cdots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & \left(\frac{\partial y_{NR}}{\partial u_{NR}}\right)^2 + \left(\frac{\partial y_{NI}}{\partial u_{NR}}\right)^2
\end{bmatrix} = \\
&= \prod_{i=1}^{N}\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right]
\end{aligned}
\tag{C.15}
$$

Using (C.15) the final expression of (C.11) is:

$$
\det(\mathbf{J}) = \det\left(\tilde{\mathbf{W}}\right) \cdot \prod_{i=1}^{N}\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right] \tag{C.16}
$$

Adaptation of W matrix

Maximizing the joint entropy (C.6) as proposed by Bell e Sejnowski [15], can be rewritten as follows:

$$
\begin{aligned}
&\max_{\mathbf{W}}\left(\ln\left(\det\left(\tilde{\mathbf{W}}\right) \cdot \prod_{i=1}^{N}\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right]\right)\right) = \\
&\max_{\mathbf{W}}\left(\ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) + \sum_{i=1}^{N}\ln\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right]\right)
\end{aligned}
\tag{C.17}
$$

Hence we calculate the increment $\Delta\mathbf{W}$ of the demixing matrix, deriving (C.17) with respect to the mixing matrix $\mathbf{W}$. It is possible to split $\Delta\mathbf{W}$ as $\Delta\mathbf{W^I} + \mathbf{\Delta W^{II}}$. So we can write

$$\begin{aligned}&\Delta\mathbf{W} = \Delta\mathbf{W}^I + \Delta\mathbf{W}^{II} \propto \\ &\propto \left(\sum_{i=1}^{N} \frac{\partial}{\partial\mathbf{W}} \ln\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right]\right) + \left(\frac{\partial}{\partial\mathbf{W}} \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right)\right)\end{aligned} \tag{C.18}$$

We first calculate:

$$\Delta\mathbf{W}^I \propto \sum_{i=1}^{N} \frac{\partial}{\partial\mathbf{W}} \ln\left[\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right] \tag{C.19}$$

We do the derivatives in (C.19):

$$\begin{aligned}&\tfrac{\partial}{\partial\mathbf{W}} \ln\left[\left(\tfrac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\tfrac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right] = \\ &= \tfrac{1}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2} \tfrac{\partial}{\partial\mathbf{W}} \left(\left(\tfrac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\tfrac{\partial y_{iI}}{\partial u_{iR}}\right)^2\right) = \\ &= \tfrac{2}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iI}}{\partial u_{iR}}\right)^2} \left(\tfrac{\partial y_{iR}}{\partial u_{iR}} \tfrac{\partial}{\partial\mathbf{W}} \tfrac{\partial y_{iR}}{\partial u_{iR}} + \tfrac{\partial y_{iI}}{\partial u_{iR}} \tfrac{\partial}{\partial\mathbf{W}} \tfrac{\partial y_{iI}}{\partial u_{iR}}\right) = \\ &= \tfrac{2}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(-\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} \left(\tfrac{\partial y_{iR}}{\partial u_{iR}} \tfrac{\partial}{\partial\mathbf{W}} \tfrac{\partial y_{iR}}{\partial u_{iR}} + \left(-\tfrac{\partial y_{iR}}{\partial u_{iI}}\right) \tfrac{\partial}{\partial\mathbf{W}} \left(-\tfrac{\partial y_{iR}}{\partial u_{iI}}\right)\right) = \\ &= \tfrac{2}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} \left(\tfrac{\partial y_{iR}}{\partial u_{iR}} \tfrac{\partial}{\partial\mathbf{W}} \tfrac{\partial y_{iR}}{\partial u_{iR}} + \tfrac{\partial y_{iR}}{\partial u_{iI}} \tfrac{\partial}{\partial\mathbf{W}} \tfrac{\partial y_{iR}}{\partial u_{iI}}\right)\end{aligned} \tag{C.20}$$

We note that $\Delta\mathbf{W^I}$ is a complex matrix so we can split it in the real and imaginary part:

$$\Delta\mathbf{W}^I = \Delta\mathbf{W}_R^I + j\Delta\mathbf{W}_I^I \tag{C.21}$$

and we can calculate these two terms separately.

Hence we have:

$$\Delta\mathbf{W}_R^I \propto \sum_{i=1}^{N} \frac{\partial}{\partial\mathbf{W}_R} \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right) \tag{C.22}$$

The elements of the matrix (C.22) are calculated as follows:

$$\begin{aligned}
&\frac{\partial}{\partial w_{ijR}} \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right) = 2\frac{\left(\frac{\partial y_{iR}}{\partial u_{iR}} \frac{\partial}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iR}} + \frac{\partial y_{iR}}{\partial u_{iI}} \frac{\partial}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iI}}\right)}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} = \\
&= 2\frac{\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)\left[\frac{\partial}{\partial u_{iR}} \frac{\partial u_{iR}}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iR}} + \frac{\partial}{\partial u_{iI}} \frac{\partial u_{iI}}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iR}}\right] + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)\left[\frac{\partial}{\partial u_{iI}} \frac{\partial u_{iI}}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iI}} + \frac{\partial}{\partial u_{iR}} \frac{\partial u_{iR}}{\partial w_{ijR}} \frac{\partial y_{iR}}{\partial u_{iI}}\right]\right)}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} = \\
&= 2\frac{\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)\left[\frac{\partial u_{iR}}{\partial w_{ijR}} \frac{\partial^2 y_{iR}}{\partial u_{iR}^2} + \frac{\partial u_{iI}}{\partial w_{ijR}} \frac{\partial}{\partial u_{iI}} \frac{\partial y_{iR}}{\partial u_{iR}}\right] + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)\left[\frac{\partial u_{iI}}{\partial w_{ijR}} \frac{\partial^2 y_{iR}}{\partial u_{iI}^2} + \frac{\partial u_{iR}}{\partial w_{ijR}} \frac{\partial}{\partial u_{iR}} \frac{\partial y_{iR}}{\partial u_{iI}}\right]\right)}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} = \\
&= 2\frac{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)\left[x_{jR} \frac{\partial^2 y_{iR}}{\partial u_{iR}^2} + x_{jI} \frac{\partial}{\partial u_{iI}} \frac{\partial y_{iR}}{\partial u_{iR}}\right] + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)\left[x_{jI} \frac{\partial^2 y_{iR}}{\partial u_{iI}^2} + x_{jR} \frac{\partial}{\partial u_{iR}} \frac{\partial y_{iR}}{\partial u_{iI}}\right]}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} = \\
&= 2x_{jR}\frac{\frac{\partial y_{iR}}{\partial u_{iR}} \frac{\partial^2 y_{iR}}{\partial u_{iR}^2} + \frac{\partial y_{iR}}{\partial u_{iI}} \frac{\partial}{\partial u_{iR}} \frac{\partial y_{iR}}{\partial u_{iI}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} + 2x_{jI}\frac{\frac{\partial y_{iR}}{\partial u_{iI}} \frac{\partial^2 y_{iR}}{\partial u_{iI}^2} + \frac{\partial y_{iR}}{\partial u_{iR}} \frac{\partial}{\partial u_{iI}} \frac{\partial y_{iR}}{\partial u_{iR}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} = \\
&= x_{jR}\Psi_{iR} + 2x_{jI}\Psi_{iI}
\end{aligned} \tag{C.23}$$

where we have posed:

$$\begin{cases}
\Psi_{iR} = 2\dfrac{\frac{\partial y_{iR}}{\partial u_{iR}} \frac{\partial^2 y_{iR}}{\partial u_{iR}^2} + \frac{\partial y_{iR}}{\partial u_{iI}} \frac{\partial}{\partial u_{iR}} \frac{\partial y_{iR}}{\partial u_{iI}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2} \\
\Psi_{iI} = 2\dfrac{\frac{\partial y_{iR}}{\partial u_{iI}} \frac{\partial^2 y_{iR}}{\partial u_{iI}^2} + \frac{\partial y_{iR}}{\partial u_{iR}} \frac{\partial}{\partial u_{iI}} \frac{\partial y_{iR}}{\partial u_{iR}}}{\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2}
\end{cases} \tag{C.24}$$

Now putting together all the elements (C.23) $\frac{\partial}{\partial w_{ijR}} \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right)$, we reconstruct the matrix (C.22):

$$\begin{aligned}
&\Delta \mathbf{W}_R^I \propto \sum_{i=1}^{N} \frac{\partial}{\partial \mathbf{W}_R} \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right) = \\
&= \sum_{i=1}^{N}\left(\Psi_{iR}\begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_R^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} + \Psi_{iI}\begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_I^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix}\right) = \mathbf{\Psi}_R \mathbf{x}_R^T + \mathbf{\Psi}_I \mathbf{x}_I^T
\end{aligned} \tag{C.25}$$

where the matrices

$$\begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_R^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_I^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix}$$

are $N \times N$ matrices with zero rows except the i-th row, which is formed by the elements of the vectors $\mathbf{x_R}$ and $\mathbf{x_I}$, the real and imaginary part of the observed signals.

Similarly we calculate the matrix $\Delta \mathbf{W}_I^I$:

$$\begin{aligned} \Delta \mathbf{W}_I^I &\propto \sum_{i=1}^{N} \frac{\partial}{\partial \mathbf{W}_I} \ln \left(\left(\frac{\partial y_{iR}}{\partial u_{iR}} \right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}} \right)^2 \right) = \\ &= \sum_{i=1}^{N} \left(-\Psi i R \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_I^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} + \Psi i I \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{x}_R^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \right) = -\mathbf{\Psi}_R \mathbf{x}_I^T + \mathbf{\Psi}_I \mathbf{x}_R^T \end{aligned} \tag{C.26}$$

Rejoining these two terms we obtain (C.21):

$$\begin{aligned} &\Delta \mathbf{W}^I = \Delta \mathbf{W}_R^I + j \Delta \mathbf{W}_J^I \propto \mathbf{\Psi}_R \mathbf{x}_R^T + \mathbf{\Psi}_I \mathbf{x}_I^T + j \left(-\mathbf{\Psi}_R \mathbf{x}_I^T + \mathbf{\Psi}_I \mathbf{x}_R^T \right) = \\ &= \left(\mathbf{\Psi}_R + j \mathbf{\Psi}_I \right) \left(\mathbf{x}_R^T - j \mathbf{x}_I^T \right) = \mathbf{\Psi} \mathbf{x}^H \end{aligned} \tag{C.27}$$

where the H operator is the Hermitian, and $\mathbf{\Psi} = \mathbf{\Psi}_R + j \mathbf{\Psi}_I$.

Now we have to calculate the matrix $\Delta \mathbf{W}^{II}$:

$$\begin{aligned} &\Delta \mathbf{W}^{II} = \Delta \mathbf{W}_R^{II} + j \Delta \mathbf{W}_I^{II} \propto \frac{\partial}{\partial \tilde{\mathbf{W}}} \ln \left(\det \left(\tilde{\mathbf{W}} \right) \right) = \\ &= \left(\tilde{\mathbf{W}}^T \right)^{-1} = \begin{pmatrix} \Delta \mathbf{W}_R^{II} & -\Delta \mathbf{W}_I^{II} \\ \Delta \mathbf{W}_I^{II} & \Delta \mathbf{W}_R^{II} \end{pmatrix} \end{aligned} \tag{C.28}$$

This because [73]

$$\frac{\partial}{\partial w_{ij}} \ln(\det W) = \frac{cof(w_{ij})}{\det W} \tag{C.29}$$

where $\det(W) = \sum_j w_{ij} cof(w_{ij})$ for any row i. For a full weight matrix, remembered that the adjoint matrix, $adj(\mathbf{W})$, is the transpose of the matrix of cofactors, we use the definition of the inverse of a matrix, so from (C.29) we have:

$$\frac{\partial}{\partial W} \ln(\det W) = \frac{(adj(W))^T}{\det W} = \left[W^T \right]^{-1} \tag{C.30}$$

In the complex case we can write:

$$\begin{aligned} &\begin{pmatrix} u_R^T & u_I^T \end{pmatrix} = \begin{pmatrix} \mathbf{x}_R^T & \mathbf{x}_I^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{W}_R^T & \mathbf{W}_I^T \\ -\mathbf{W}_I^T & \mathbf{W}_R^T \end{pmatrix} \\ &\Downarrow \\ &\begin{cases} u_R^T = \mathbf{x}_R^T \mathbf{W}_R^T - \mathbf{x}_I^T \mathbf{W}_I^T \\ u_I^T = \mathbf{x}_R^T \mathbf{W}_I^T + \mathbf{x}_I^T \mathbf{W}_R^T \end{cases} \end{aligned} \tag{C.31}$$

because the complex signals u_i are obtained by the product of the observed signals $\mathbf{x}$ for the demixing matrix $\mathbf{W}$.

If we invert this linear system (C.31), we obtain:

$$\begin{array}{l}\left(\mathbf{w}^H\right)^{-1} = \\ = \begin{pmatrix} \left(\mathbf{w}_I^T\right)^{-1}\left(\mathbf{w}_R^T\left(\mathbf{w}_I^T\right)^{-1}+\mathbf{w}_I^T\left(\mathbf{w}_R^T\right)^{-1}\right)^{-1} & -\left(\mathbf{w}_R^T\right)^{-1}\left(\mathbf{w}_R^T\left(\mathbf{w}_I^T\right)^{-1}+\mathbf{w}_I^T\left(\mathbf{w}_R^T\right)^{-1}\right)^{-1} \\ \left(\mathbf{w}_R^T\right)^{-1}\left(\mathbf{w}_R^T\left(\mathbf{w}_I^T\right)^{-1}+\mathbf{w}_I^T\left(\mathbf{w}_R^T\right)^{-1}\right)^{-1} & \left(\mathbf{w}_I^T\right)^{-1}\left(\mathbf{w}_R^T\left(\mathbf{w}_I^T\right)^{-1}+\mathbf{w}_I^T\left(\mathbf{w}_R^T\right)^{-1}\right)^{-1} \end{pmatrix} = \\ = \begin{pmatrix} \Delta\mathbf{w}_R^{II} & -\Delta\mathbf{w}_I^{II} \\ \Delta\mathbf{w}_I^{II} & \Delta\mathbf{w}_R^{II} \end{pmatrix}\end{array} \tag{C.32}$$

In a compact form we can pose

$$\mathbf{W}^H = \mathbf{W}_R^T - j\mathbf{W}_I^T \tag{C.33}$$

Introducing a complex matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{A}_R + j\mathbf{A}_I \tag{C.34}$$

such that

$$\mathbf{A}\mathbf{W}^H = \mathbf{I}$$

thus we write from (C.33) and (C.34):

$$\begin{cases} \mathbf{A}_R\mathbf{W}_R^T + \mathbf{A}_I\mathbf{W}_I^T = \mathbf{I} \\ -\mathbf{A}_R\mathbf{W}_I^T + \mathbf{A}_I\mathbf{W}_R^T = \mathbf{0} \end{cases} \tag{C.35}$$

where

$$\mathbf{A}_R = \left(\mathbf{W}_I^T\right)^{-1}\left(\mathbf{W}_I^T\left(\mathbf{W}_R^T\right)^{-1} + \mathbf{W}_R^T\left(\mathbf{W}_I^T\right)^{-1}\right)^{-1} \tag{C.36}$$

and

$$\mathbf{A}_I = \left(\mathbf{W}_R^T\right)^{-1}\left(\mathbf{W}_I^T\left(\mathbf{W}_R^T\right)^{-1} + \mathbf{W}_R^T\left(\mathbf{W}_I^T\right)^{-1}\right)^{-1} \tag{C.37}$$

Hence from (C.36), (C.37) and the solution of (C.31), $\Delta\mathbf{W}^{II}$ can be written as follows

$$\Delta\mathbf{W}^{II} \propto \frac{\partial}{\partial\mathbf{W}} \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) = \left(\mathbf{W}^H\right)^{-1} \tag{C.38}$$

Finally, the expression for the adaptation (C.18) of the demixing matrix is derived joining (C.27) and (C.38):

$$\Delta\mathbf{W} = \Delta\mathbf{W}^I + \Delta\mathbf{W}^{II} \propto \mathbf{\Psi}\mathbf{x}^H + \left(\mathbf{W}^H\right)^{-1} \tag{C.39}$$

This expression proves the algorithm (7.17).

Adaptation of control points

To adapt the control points of the spline functions, we have to use the ME approach (7.11), maximizing the joint entropy [55] of the output vector $H(\mathbf{y})$ with respect to each control point

$$\begin{array}{l}\Delta Q_{j,i_R+m_R,i_I+m_I} \propto \frac{\partial H(\mathbf{y})}{\partial Q_{j,i_R+m_R,i_I+m_I}} = \\ = \frac{\partial E\{\ln(\det(\mathbf{J}))\}}{\partial Q_{j,i_R+m_R,i_I+m_I}} + \frac{\partial H(\mathbf{x})}{\partial Q_{j,i_R+m_R,i_I+m_I}}\end{array} \tag{C.40}$$

where $m_R, m_I = 0, \ldots, 3$. We have

$$\frac{\partial H(\mathbf{x})}{\partial Q_{j,i_R+m_R,i_I+m_I}} = 0$$

because $H(\mathbf{x})$ does not depend by the control points of spline functions. Then we can do the following reduction:

$$\frac{\partial E\left\{\ln\left(\det\left(\mathbf{J}\right)\right)\right\}}{\partial Q_{j,i_R+m_R,i_I+m_I}} \approx \frac{\partial \ln\left(\det\left(\mathbf{J}\right)\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} \tag{C.41}$$

accepting a small error, because we have not the probability density function of the quantity in the braces.

Applying (C.41), the adaptation algorithm of the control points (C.40) is expressed as

$$\Delta Q_{j,i_R+m_R,i_I+m_I} \propto \frac{\partial \ln\left(\det\left(\mathbf{J}\right)\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} = \frac{1}{\det\left(\mathbf{J}\right)} \frac{\partial \det\left(\mathbf{J}\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} \tag{C.42}$$

Now using (C.16) we have

$$\begin{aligned} &\ln\left(\det\left(\mathbf{J}\right)\right) = \ln\left(\det\left(\tilde{\mathbf{W}}\right) \cdot \prod_{i=1}^{N}\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right)\right) = \\ &= \sum_{i=1}^{N} \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right) + \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) \end{aligned} \tag{C.43}$$

Because $\mathbf{W}$ does not depend by $\mathbf{Q}$ we have that

$$\frac{\partial \det\left(\tilde{\mathbf{W}}\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} = 0$$

For the linearity property, we can exchange the summation in (C.43) with the derivatives, thus

$$\begin{aligned} &\frac{\partial \ln\left(\left(\frac{\partial y_{iR}}{\partial u_{iR}}\right)^2 + \left(\frac{\partial y_{iR}}{\partial u_{iI}}\right)^2\right)}{\partial Q_{j,i_R+m_R,i_I+m_I}} = \\ &= \begin{cases} 0 & i \neq j \\ 2\dfrac{\frac{\partial y_{jR}}{\partial u_{jR}} \frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}} \frac{\partial y_{jR}}{\partial u_{jR}} + \frac{\partial y_{jR}}{\partial u_{jJ}} \frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}} \frac{\partial y_{jR}}{\partial u_{jI}}}{\left(\frac{\partial y_{jR}}{\partial u_{jR}}\right)^2 + \left(\frac{\partial y_{jR}}{\partial u_{jI}}\right)^2} & i = j \end{cases} \end{aligned} \tag{C.44}$$

where $i \neq j$ is the case of derivatives between independent variables.

Using the spline approximation in matrix notation, we calculate the derivatives in (C.44) as the derivatives of the bi-dimensional spline functions with respect to the real and the imaginary input (remembering that is $\mathbf{T}_{\nu,1}(\nu_R)$ and $\mathbf{T}_{\nu,2}(\nu_I)$):

$$\begin{cases} \frac{\partial y_{iR}}{\partial u_{iR}} = \frac{\partial y_{iR}}{\partial \nu_{iR}} \frac{\partial \nu_{iR}}{\partial u_{iR}} = \frac{1}{\Delta \nu_{iR}} \mathbf{T}_{\nu,2i} \cdot \mathbf{M} \cdot \left(\dot{\mathbf{T}}_{\nu,1i} \cdot \mathbf{M} \cdot \mathbf{Q}_{iR}\right)^T \\ \frac{\partial y_{iR}}{\partial u_{iI}} = \frac{\partial y_{iR}}{\partial \nu_{iI}} \frac{\partial \nu_{iI}}{\partial u_{iI}} = \frac{1}{\Delta \nu_{iI}} \dot{\mathbf{T}}_{\nu,2i} \cdot \mathbf{M} \cdot \left(\mathbf{T}_{\nu,1i} \cdot \mathbf{M} \cdot \mathbf{Q}_{iR}\right)^T \end{cases} \tag{C.45}$$

where $\Delta\nu_R$ and $\Delta\nu_J$ are the sample steps of the spline functions.

So using (C.45) the terms in expression (C.44) become

$$\begin{aligned}&\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial u_{jR}}=\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial \nu_{jR}}\frac{\partial \nu_{jR}}{\partial u_{jR}}=\\&=\frac{1}{\Delta\nu_{jR}}\frac{\partial \mathbf{T}_{\nu,2i}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu,1i}\cdot\mathbf{M}\cdot\mathbf{Q}_{iR}\right)^T}{\partial Q_{j,i_R+m_R,i_I+m_I}}=\frac{1}{\Delta\nu_{jR}}\mathbf{T}_{\nu,2i}\cdot\mathbf{M}_{m_I}\cdot\left(\dot{\mathbf{T}}_{\nu,1i}\cdot\mathbf{M}_{m_R}\right)^T\end{aligned} \tag{C.46}$$

and

$$\begin{aligned}&\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial u_{jI}}=\frac{\partial}{\partial Q_{j,i_R+m_R,i_I+m_I}}\frac{\partial y_{jR}}{\partial \nu_{jI}}\frac{\partial \nu_{jI}}{\partial u_{jI}}=\\&=\frac{1}{\Delta\nu_{jI}}\frac{\partial \dot{\mathbf{T}}_{\nu,2i}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu,1i}\cdot\mathbf{M}\cdot\mathbf{Q}_{iR}\right)^T}{\partial Q_{j,i_R+m_R,i_I+m_I}}=\frac{1}{\Delta\nu_{jI}}\dot{\mathbf{T}}_{\nu,2i}\cdot\mathbf{M}_{m_I}\cdot\left(\mathbf{T}_{\nu,1i}\cdot\mathbf{M}_{m_R}\right)^T\end{aligned} \tag{C.47}$$

where $\mathbf{M_k}$ is a matrix in which all the elements are zero, except the k-th column, which is equal to the k-th column of $\mathbf{M}$.

Introducing (C.46) and (C.47) in the expression (C.44), and choosing $\Delta\nu_{jR}=\Delta\nu_{jJ}$ for simplicity, the adaptation equation of the control points (C.40) is written as follows:

$$\begin{aligned}\Delta Q_{j,i_R+m_R,i_I+m_I}=2\eta_{\mathbf{Q}}\Bigg(&\frac{\left(\mathbf{T}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)\left(\mathbf{T}_{\nu,2j}\cdot\mathbf{M}_{m_I}\cdot\left(\dot{\mathbf{T}}_{\nu,1j}\cdot\mathbf{M}_{m_R}\right)^T\right)}{\left(\mathbf{T}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)^2}+\\&+\frac{\left(\dot{\mathbf{T}}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)\left(\dot{\mathbf{T}}_{\nu,2j}\cdot\mathbf{M}_{m_I}\cdot\left(\mathbf{T}_{\nu,1j}\cdot\mathbf{M}_{m_R}\right)^T\right)}{\left(\mathbf{T}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)^2+\left(\dot{\mathbf{T}}_{\nu,2j}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu,1j}\cdot\mathbf{M}\cdot\mathbf{Q}_{jR}\right)^T\right)^2}\Bigg)\end{aligned} \tag{C.48}$$

where η_Q is the learning rate, a real and positive constant.

The expressions (C.44) and (C.48) prove the algorithms (7.22) and (7.23).

C.2 Proof of Theorem 17

See also [212].

Sufficient condition: existence of the solution.

Given the channel model (4.3), it is easy to verify that if $\mathbf{s}[n]$ is a spatially independent complex random vector, under the given assumptions, $\mathbf{u}[n]$ will be spatially independent too because the channel does not produce any mixing. Given the mixing model $\mathcal{F}(\mathbf{A},\mathbf{F})$, assumptions a), b), and c) guarantee that there exists a matrix $\mathbf{W}$ and N functions $g_i(\bullet)$ such that:

$$\begin{aligned}&\mathbf{G}\left\{\mathbf{F}\left\{\mathbf{A}\mathbf{s}\left[n\right]\right\}\right\}=\mathbf{A}\mathbf{s}\left[n\right]\\&\mathbf{W}\mathbf{A}\mathbf{s}\left[n\right]=\mathbf{P}\mathbf{\Lambda}\mathbf{s}\left[n\right]\end{aligned} \tag{C.49}$$

Based on (C.49), the input-output transformation can be written as

$$\mathbf{W}\mathbf{G}\left[\mathbf{F}\left\{\mathbf{A}\mathbf{s}\left[n\right]\right\}\right]=\mathbf{P}\mathbf{\Lambda}\mathbf{s}\left[n\right]$$

Necessary condition: uniqueness of the solution.

This condition proves that if $\mathbf{u}[n]$ is a spatially independent random vector, the channel model must be (4.3).

The complex transformation which maps $\mathbf{s}$ into $\mathbf{y}$ is:

$$\begin{aligned}
&\mathbf{y}[n] = \mathbf{WG}\left[\mathbf{F}\left\{\mathbf{As}[n]\right\}\right] = \\
&= \mathbf{W}\begin{bmatrix} g_{R1}\left(f_{R1}\left(\sum_{i=1}^{N} a_{1i}s_i[n]\right)\right) + jg_{I1}\left(f_{I1}\left(\sum_{i=1}^{N} a_{1i}s_i[n]\right)\right) \\ \vdots \\ g_{RN}\left(f_{RN}\left(\sum_{i=1}^{N} a_{Ni}s_i[n]\right)\right) + jg_{IN}\left(f_{IN}\left(\sum_{i=1}^{N} a_{Ni}s_i[n]\right)\right) \end{bmatrix}
\end{aligned} \tag{C.50}$$

It is possible to rewrite the complex map (C.50) in a real form as follows:

$$\begin{aligned}
&\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_R[n] \\ \mathbf{u}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \mathrm{G}_R\{\mathbf{x}_R[n]\} \\ \mathrm{G}_I\{\mathbf{x}_I[n]\} \end{bmatrix} \\
&\begin{bmatrix} \mathbf{x}_R[n] \\ \mathbf{x}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{F}_R\{\mathbf{v}_R[n]\} \\ \mathbf{F}_I\{\mathbf{v}_I[n]\} \end{bmatrix} \\
&\tilde{\mathbf{v}}[n] = \begin{bmatrix} \mathbf{v}_R[n] \\ \mathbf{v}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix} \begin{bmatrix} \mathbf{s}_R[n] \\ \mathbf{s}_I[n] \end{bmatrix} = \tilde{\mathbf{A}}\tilde{\mathbf{s}}
\end{aligned} \tag{C.51}$$

in which $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix}$ and $\tilde{\mathbf{W}} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix}$ are $2N \times 2N$ matrices. It is important to underline that in (C.51) there are only real elements. For assumption (a) matrix $\mathbf{A}$ is non singular, then due to its structure it is evident that $\tilde{\mathbf{A}}$ has to be non singular too.

The pdf of $\tilde{\mathrm{s}}$ can be written as a function of the pdf of $\tilde{\mathbf{u}}$:

$$\begin{aligned}
&p_{\tilde{\mathrm{s}}}(\tilde{\mathrm{s}}) = \prod_{i=1}^{N} p_{s_{Ri}}(s_{Ri})\, p_{s_{Ii}}(s_{Ii}) = \\
&= \prod_{i=1}^{N} p_{Ru_i}\left(\begin{array}{l} \sum_{j=1}^{N} w_{Rij}g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm} - \sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right] + \\ -\sum_{j=1}^{N} w_{Iij}g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm} + \sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right] \end{array}\right) p_{u_{Ii}}(y_{Ii})\left|\tilde{\mathrm{J}}\right| \qquad \forall \tilde{\mathrm{s}} \in \mathbb{R}^{2N}
\end{aligned} \tag{C.52}$$

in which $\tilde{\mathbf{J}}$ is the Jacobian matrix of the application which maps $\tilde{\mathbf{s}}$ into $\tilde{\mathbf{y}}$.

From assumption (d)$\exists \bar{\mathbf{s}} \in \mathbb{C}^N$ such that $p_{\mathbf{s}}(\bar{\mathbf{s}}) \equiv 0$. Then considering $\tilde{\mathbf{s}}[n] = [\mathbf{s}_R[n], \mathbf{s}_I[n]]^T$ in which $\mathbf{s}_R[n], \mathbf{s}_I[n] \in \mathbb{R}$ the assumption (d) can be reformulated as follows: $\exists \bar{\tilde{\mathbf{s}}} \in \mathbb{R}^{2N}\left|p_{\tilde{\mathbf{s}}}(\bar{\tilde{\mathbf{s}}}) \equiv 0\right.$.

From (C.50), for a non null Jacobian $\tilde{\mathbf{J}}$, there exists some $\tilde{\mathbf{u}}^0 = [u^0_{R1}, \ldots, u^0_{RN}, u^0_{I1}, \ldots u^0_{IN}] \in \mathbb{R}^{2N}$ such that $\prod_{i=1}^{N} p_{u_{Ri}}(u^0_{Ri})\, p_{u_{Ii}}(u^0_{Ii}) = 0$. Consequently there exists at least one integer i such that $p_{u_{Ri}}(u^0_{Ri}) = 0$ or $p_{u_{Ii}}(u^0_{Ii}) = 0$.

This leads to the following equation:

$$\begin{aligned}\tilde{u}_i^0 = &\sum_{j=1}^{N} w_{Rij} g_{Rj} \left[f_{Rj} \left(\sum_{m=1}^{N} a_{Rjm} s_{Rm} - \sum_{m=1}^{N} a_{Ijm} s_{Im} \right) \right] + \\ &- \sum_{j=1}^{N} w_{Iij} g_{Ij} \left[f_{Ij} \left(\sum_{m=1}^{N} a_{Rjm} s_{Im} + \sum_{m=1}^{N} a_{Ijm} s_{Rm} \right) \right]\end{aligned} \tag{C.53}$$

in which $\tilde{u}_i^0$ is the i-th element of the vector $\tilde{\mathbf{u}}^0$. Solutions of (C.53) lie on $\mathcal{H}_i(\tilde{\mathbf{s}})$, which is a hyper-surface in $\mathbb{R}^{2N}$. It is evident that $\forall \tilde{\mathbf{s}} \in H_i(\tilde{\mathbf{s}}) \Rightarrow p_{\tilde{\mathbf{s}}}(\tilde{\mathbf{s}}) = 0$. For a given i, $\mathcal{H}_i(\tilde{\mathbf{s}})$ is parallel to the hyperplane orthogonal to the axis $\tilde{s}_i$ (considering as $\tilde{s}_i$ the i-th element of the vector $\tilde{\mathbf{s}}$). Suppose that $\mathcal{H}_i(\tilde{\mathbf{s}})$ is not parallel to any $\tilde{s}_i = 0$ plane. The projection of $\mathcal{H}_i(\tilde{\mathbf{s}})$ onto $\tilde{s}_i = 0$ should be $\mathbb{R}$: $\forall \tilde{s}_i \in \mathbb{R} \exists \tilde{s}_1, \ldots, \tilde{s}_{i-1}, \tilde{s}_{i+1}, \ldots, \tilde{s}_N, \ldots, \tilde{s}_{2N} : \tilde{\mathbf{s}} \in H_i \Rightarrow p_{\tilde{\mathbf{s}}}(\tilde{\mathbf{s}}) \equiv 0$. This cannot be true since $\int_S p_{\tilde{s}}(\tilde{\mathrm{s}})\, d\tilde{s} = 1$. Without loss of generality, it can be noted that:

$$\begin{aligned}&\sum_{j=1}^{N} w_{Rij} g_{Rj} \left[f_{Rj} \left(\sum_{m=1}^{N} a_{Rjm} s_{Rm} - \sum_{m=1}^{N} a_{Ijm} s_{Im} \right) \right] + \\ &- \sum_{j=1}^{N} w_{Iij} g_{Ij} \left[f_{Ij} \left(\sum_{m=1}^{N} a_{Ijm} s_{Rm} + \sum_{m=1}^{N} a_{Rjm} s_{Im} \right) \right] = b_{\sigma(i)} \left(\tilde{s}_{\sigma(i)} \right), \quad i = 1, \ldots, N \\ &\sum_{j=1}^{N} w_{Iij} g_{Rj} \left[f_{Rj} \left(\sum_{m=1}^{N} a_{Rjm} s_{Rm} - \sum_{m=1}^{N} a_{Ijm} s_{Im} \right) \right] + \\ &+ \sum_{j=1}^{N} w_{Rij} g_{Ij} \left[f_{Ij} \left(\sum_{m=1}^{N} a_{Ijm} s_{Rm} + \sum_{m=1}^{N} a_{Rjm} s_{Im} \right) \right] = b_{\sigma(i)} \left(\tilde{s}_{\sigma(i)} \right), \; i = N+1, \ldots, 2N\end{aligned} \tag{C.54}$$

where $b_{\sigma(i)}\left(\tilde{s}_{\sigma(i)}\right)$ is a generic function depending only on $\tilde{s}_{\sigma(i)}$ (that is the source for the i-th output). Then without any loss of generality taking $\sigma(i) = i$

$$\begin{aligned}&\sum_{j=1}^{N} w_{Rij} g_{Rj} \left[f_{Rj} \left(\sum_{m=1}^{N} a_{Rjm} s_{Rm} - \sum_{m=1}^{N} a_{Ijm} s_{Im} \right) \right] + \\ &- \sum_{j=1}^{N} w_{Iij} g_{Ij} \left[f_{Ij} \left(\sum_{m=1}^{N} a_{Ijm} s_{Rm} - \sum_{m=1}^{N} a_{Rjm} s_{Im} \right) \right] = b_i \left(\tilde{s}_i \right), \quad i = 1, \ldots, N \\ &\sum_{j=1}^{N} w_{Iij} g_{Rj} \left[f_{Rj} \left(\sum_{m=1}^{N} a_{Rjm} s_{Rm} - \sum_{m=1}^{N} a_{Ijm} s_{Im} \right) \right] + \\ &+ \sum_{j=1}^{N} w_{Rij} g_{Ij} \left[f_{Ij} \left(\sum_{m=1}^{N} a_{Ijm} s_{Rm} - \sum_{m=1}^{N} a_{Rjm} s_{Im} \right) \right] = b_i \left(\tilde{s}_i \right), \; i = N+1, \ldots, 2N\end{aligned} \tag{C.55}$$

Derivation with respect to s yields

$$\begin{aligned}\begin{bmatrix} \dot{b}_1(s_1) & 0 \\ 0 & \dot{b}_{2N}(\tilde{s}_{2N}) \end{bmatrix} = \tilde{\mathrm{W}} &\begin{bmatrix} \dot{f}_{R1}\left[f_{R1}\left(\mathbf{v}_R(\tilde{\mathbf{s}})\right)\right] & 0 \\ 0 & \dot{g}_{IN}\left[\left(\mathbf{x}_I(\tilde{\mathbf{s}})\right)\right] \end{bmatrix} \cdot \\ &\begin{bmatrix} \dot{f}_{R1}\left(\mathbf{v}_R(\tilde{\mathbf{s}})\right) & 0 \\ 0 & \dot{g}_{IN}\left(\mathbf{v}_I(\tilde{\mathbf{s}})\right) \end{bmatrix} \tilde{\mathrm{A}}\end{aligned} \tag{C.56}$$

Considering $\tilde{\mathbf{s}}_1$ and $\tilde{\mathbf{s}}_2$ as coordinates of the hypersurface $\mathcal{H}(\tilde{\mathbf{s}})$, (C.56) can be evaluated in $\tilde{\mathbf{s}}_1$ and $\tilde{\mathbf{s}}_2$ in as follows:

$$\left\{ \begin{array}{l} \mathbf{D}\left(\tilde{\mathbf{s}}_1\right) = \tilde{\mathbf{W}} \mathbf{\Lambda}_{\dot{\mathbf{G}}}\left(\tilde{\mathbf{s}}_1\right) \mathbf{\Lambda}_{\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_1\right) \tilde{\mathbf{A}} \\ \mathbf{D}\left(\tilde{\mathbf{s}}_2\right) = \tilde{\mathbf{W}} \mathbf{\Lambda}_{\dot{\mathbf{G}}}\left(\tilde{\mathbf{s}}_2\right) \mathbf{\Lambda}_{\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_2\right) \tilde{\mathbf{A}} \end{array} \right. \rightarrow \left\{ \begin{array}{l} \mathbf{D}\left(\mathbf{s}_1\right) = \tilde{\mathbf{W}} \mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_1\right) \tilde{\mathbf{A}} \\ \mathbf{D}\left(\mathbf{s}_2\right) = \tilde{\mathbf{W}} \mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_2\right) \tilde{\mathbf{A}} \end{array} \right. \tag{C.57}$$

Then eliminating $\tilde{\mathbf{W}}$:

$$\tilde{\mathrm{A}} \underbrace{\begin{bmatrix} d_{11}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) & & 0 \\ & \ddots & \\ 0 & & d_{2N2N}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) \end{bmatrix}^{-1} \mathrm{D}\left(\tilde{\mathbf{s}}_2\right) \mathrm{D}\left(\tilde{\mathbf{s}}_1\right)}_{} = \underbrace{\begin{bmatrix} \lambda_{11}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) & & 0 \\ & \ddots & \\ 0 & & \lambda_{2N2N}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) \end{bmatrix}^{-1} \Lambda_{\dot{\mathrm{F}}\dot{\mathrm{G}}}\left(\tilde{\mathbf{s}}_2\right) \Lambda_{\dot{\mathrm{F}}\dot{\mathrm{G}}}\left(\tilde{\mathbf{s}}_1\right)}_{} \tilde{\mathrm{A}}$$

.

As $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix}$ is regular and non singular, for each pair of non zero elements of $\tilde{\mathbf{A}}$ it is possible to write:

$$\left\{ \begin{array}{l} \tilde{a}_{ij}\left[d_{jj}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) - \lambda_{ii}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right)\right] = 0 \\ \tilde{a}_{hj}\left[d_{jj}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) - \lambda_{hh}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right)\right] = 0 \end{array} \right. \Rightarrow \lambda_{ii}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) = \lambda_{hh}\left(\tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1\right) \forall \tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1 \in H \tag{C.58}$$

in which $\tilde{a}_{ij}$ for $i, j = 1, \ldots, 2N$ is an element of $\tilde{\mathbf{A}}$. From (C.58) it follows:

$$\frac{\dot{\tilde{g}}_i\left[\hat{\tilde{f}}_i\left(\left(\tilde{\mathbf{A}}\right)_{\sigma(i)} \tilde{\mathbf{s}}_1\right)\right] \dot{\hat{\tilde{f}}}_i\left(\left(\tilde{\mathbf{A}}\right)_i \tilde{\mathbf{s}}_1\right)}{\dot{\tilde{g}}_h\left[\hat{\tilde{f}}_h\left(\left(\tilde{\mathbf{A}}\right)_h \tilde{\mathbf{s}}_2\right)\right] \dot{\hat{\tilde{f}}}_h\left(\left(\tilde{\mathbf{A}}\right)_h \tilde{\mathbf{s}}_2\right)} = C, \quad i = 1, \ldots, 2N, \ \forall \tilde{\mathbf{s}}_2, \tilde{\mathbf{s}}_1 \tag{C.59}$$

where $\tilde{g}_i$ is the i-th element of the vector $[\mathbf{G}_R, \mathbf{G}_I]^T$, $\hat{\tilde{f}}_i\left(\left(\tilde{\mathbf{A}}\right)_i \tilde{\mathbf{s}}_1\right) = \alpha \tilde{f}_i\left(\left(\tilde{\mathbf{A}}\right)_i \tilde{\mathbf{s}}_1\right)$ and C is a constant. For the two linear forms in (C.58) $\left(\tilde{\mathrm{A}}\right)_i \mathrm{s}_1 \quad \left(\tilde{\mathbf{A}}\right)_h \mathrm{s}_2$ are independent as assumed in (a), it is possible to express the (C.58) in the following way: $\dot{\tilde{g}}_i\left[\hat{\tilde{f}}_i(x)\right] \dot{\hat{\tilde{f}}}_i(x) = C \dot{\tilde{g}}_h\left[\hat{\tilde{f}}_h(y)\right] \dot{\hat{\tilde{f}}}_h(y), \ \forall x, y \in \mathbb{R}$. This can be true if and only if $\tilde{g}_i(\cdot)$ is the inverse of $\tilde{f}_i(\cdot)$, up to a scaling factor.

From (C.51), by previous results it follows:

$$\tilde{\mathbf{u}}[n] = \tilde{\mathbf{W}} \begin{bmatrix} \xi_1 & & \\ & \ddots & \\ & & \xi_{2N} \end{bmatrix} \tilde{\mathbf{A}} \tilde{\mathbf{s}}[n] \tag{C.60}$$

where $\xi_1, \ldots, \xi_N$ are scaling coefficients.

This method reduces the mixing-demixing non linear channel (C.49) to the simpler, linear model (C.60). By considering:

$$\begin{aligned}\tilde{\mathbf{W}}' &= \tilde{\mathbf{W}} \begin{bmatrix} \xi_1 & & \\ & \ddots & \\ & & \xi_N \end{bmatrix} \\ \tilde{\mathbf{u}}[n] &= \tilde{\mathbf{W}} \begin{bmatrix} \xi_1 & & \\ & \ddots & \\ & & \xi_{2N} \end{bmatrix} \tilde{\mathbf{A}}\tilde{\mathbf{s}}[n]\end{aligned} \tag{C.61}$$

(C.60) can be transformed into $\tilde{\mathbf{u}} = \tilde{\mathbf{W}}'\tilde{\mathbf{A}}\tilde{\mathbf{s}}$. For this formulation independent outputs can be obtained if and only if

$$\tilde{\mathbf{W}}'\tilde{\mathbf{A}} = \tilde{\mathbf{P}} \begin{bmatrix} \varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_{2N} \end{bmatrix} \tag{C.62}$$

where $\tilde{\mathbf{P}}$ is a $2N \times 2N$ permutation matrix.

Considering the structure of $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix}$ and $\tilde{\mathbf{W}}' = \begin{bmatrix} \mathbf{W}_R\mathbf{D}_1 & -\mathbf{W}_I\mathbf{D}_1 \\ \mathbf{W}_I\mathbf{D}_2 & \mathbf{W}_R\mathbf{D}_2 \end{bmatrix}$ in which $\mathbf{D}_1 = \begin{bmatrix} \xi_1 & 0 \\ 0 & \xi_N \end{bmatrix}$ and $\mathbf{D}_2 = \begin{bmatrix} \xi_{N+1} & 0 \\ 0 & \xi_{2N} \end{bmatrix}$ are $N \times N$ matrices, it is possible to rewrite (C.62) in the following way:

$$\begin{bmatrix} \mathbf{W}_R\mathbf{D}_1 & -\mathbf{W}_I\mathbf{D}_1 \\ \mathbf{W}_I\mathbf{D}_2 & \mathbf{W}_R\mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix} = \begin{bmatrix} \mathbf{P}_R & -\mathbf{P}_I \\ \mathbf{P}_I & \mathbf{P}_R \end{bmatrix} \begin{bmatrix} \varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_{2N} \end{bmatrix} \tag{C.63}$$

in which the permutation matrix $\tilde{\mathbf{P}}$ has four blocks, $\mathbf{P}_R$ and $\mathbf{P}_I$ are themselves permutation matrix of dimension $N \times N$.
For the independence of output signals the generic element p_{ij} of the matrix $\tilde{\mathbf{P}}$ is such that $\begin{cases} p_{i,j} \neq 0 \Leftrightarrow p_{i+N,j+N} \neq 0 \\ p_{i,j+N} \neq 0 \Leftrightarrow p_{i+N,j} \neq 0 \end{cases}$ $i, j \in 1, \ldots, N$.
With simple considerations eq. (C.63) proves the Theorem 17.

C.3 The Nonlinear Case: ME Algorithm

This section presents the derivation of the learning rule for the complex algorithm (see also [212]). We define the following symbols (see Fig. 4.2 and Fig. 7.7):

$\mathbf{s} = \mathbf{s}_R + j\mathbf{s}_I$: independent input sources in the complex domain;

$\mathbf{x} = \mathbf{x}_R + j\mathbf{x}_I$: mixed signals, input of the neural network for separation;

$\mathbf{u} = \mathbf{u}_R + j\mathbf{u}_I$: estimate of the input sources;

$\mathbf{y} = \mathbf{y}_R + j\mathbf{y}_I$: neural network outputs;

$\mathbf{W} = \mathbf{W}_R + j\mathbf{W}_I$: neural network weights.

Let $\mathbf{u} = \mathbf{u}_R + j\mathbf{u}_I$ be a vector of N complex random variables (c.r.v.). Its pdf can be indicated as $p_{\mathbf{u}}(u_{R1}, \ldots, u_{RN}, u_{I1}, \ldots, u_{IN})$. The subscripts R and I indicate the real and imaginary part respectively.

The N c.r.vs. are statistically independent iff their joint pdf can be expressed as

$$p_{\mathbf{u}}(u_{R1}, \ldots, u_{RN}, u_{I1}, \ldots, u_{IN}) = \prod_{i=1}^{N} p_{u_i}(u_{Ri}, u_{Ii}) \tag{C.64}$$

The nonlinear complex activation function used in this algorithm is based on the use of flexible spline neurons. They are used to introduce an upper bound for the entropy of output signals which is naturally unbounded. Let $\mathbf{y} = \mathbf{y}_R + j\mathbf{y}_I$ such that, using the splitting model (5.3):

$$y_i = y_{Ri} + jy_{Ii} = h_i^R(u_{Ri}) + jh_i^I(u_{Ii}) \tag{C.65}$$

be the c.r.v. at the output of the activation function, then it is possible to consider the real variable $\tilde{\mathbf{y}}$ defined as $\tilde{\mathbf{y}} \triangleq [\mathbf{y}_R, \mathbf{y}_I]^T$.

In this way it is also possible to express the pdf of $\tilde{\mathbf{y}}$ as a function of the pdf of $\tilde{\mathbf{x}}$: $p_{\tilde{\mathbf{y}}}\left(\begin{bmatrix} \mathbf{y}_{\mathrm{R}} \\ \mathbf{y}_{\mathrm{I}} \end{bmatrix}\right) = \frac{p_{\tilde{\mathbf{x}}}\left(\begin{bmatrix} \mathbf{x}_{\mathrm{R}} \\ \mathbf{x}_{\mathrm{I}} \end{bmatrix}\right)}{\det(\tilde{\mathbf{J}})}$, being $\tilde{\mathbf{J}}$ the Jacobian matrix of the nonlinear application:

$$\begin{aligned} \tilde{\mathbf{y}} &= \begin{bmatrix} \mathbf{y}_R[n] \\ \mathbf{y}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{h}^R\{\mathbf{u}_R[n]\} \\ \mathbf{h}^I\{\mathbf{u}_I[n]\} \end{bmatrix} \\ \tilde{\mathbf{u}}[n] &= \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \mathbf{G}_R\{\mathbf{x}_R[n]\} \\ \mathbf{G}_I\{\mathbf{x}_I[n]\} \end{bmatrix} = \tilde{\mathbf{W}}\mathbf{G}[\tilde{\mathbf{x}}] \end{aligned} \tag{C.66}$$

The joint entropy of $\tilde{\mathbf{y}}$ can be defined as:

$$H(\tilde{\mathbf{y}}) = -\int_{-\infty}^{+\infty} p_{\tilde{\mathbf{y}}}(\mathbf{y}_R, \mathbf{y}_I) \ln\left(p_{\tilde{\mathbf{y}}}(\mathbf{y}_R, \mathbf{y}_I)\right) d\mathbf{y}_R d\mathbf{y}_I \tag{C.67}$$

According to the INFOMAX principle, separation can be obtained by maximizing the output entropy $H(\tilde{\mathbf{y}})$ [55] of the network represented in Fig. 7.7 with respect to the network's weights, i.e. the elements of the set $\mathbf{\Phi} = \left\{\mathbf{W}, \mathbf{Q}_{\mathrm{G}}^{NL}, \mathbf{Q}_h^{SC}\right\}$. $\mathbf{Q}_{\mathrm{G}}^{NL} = \left\{\mathbf{Q}_{\mathrm{Re(G)}}^{NL}, \mathbf{Q}_{\mathrm{Im(G)}}^{NL}\right\}$ and $\mathbf{Q}_h^{SC} = \left\{\mathbf{Q}_{h^R}^{SC}, \mathbf{Q}_{h^I}^{SC}\right\}$ are the control points of the spline neurons; in particular $\mathbf{Q}_{\mathrm{Re}(.)}$ and $\mathbf{Q}_{\mathrm{Im}(.)}$ are

the control points of the real part and the imaginary part of the spline neurons respectively. So the cost function $\mathcal{L}\{\tilde{\mathbf{y}}, \mathbf{\Phi}[\mathbf{n}]\}$ is:

$$\mathcal{L}\{\tilde{\mathbf{y}}, \mathbf{\Phi}[n]\} = E\{-\ln(p_{\tilde{\mathbf{y}}}(y_{R1}, \ldots, y_{RN}, y_{I1}, \ldots, y_{IN}))\} \tag{C.68}$$

Maximization is performed by the stochastic gradient method

$$\mathbf{\Phi}[n+1] = \mathbf{\Phi}[n] + \eta \nabla_{\mathbf{\Phi}} L\{\mathbf{\Phi}, \tilde{\mathbf{y}}[n]\} \tag{C.69}$$

It is possible to rewrite the network's input-output relationship as follows:

$$\begin{aligned} u_{Ri} + j u_{Ii} = &\sum_{k=1}^{N} (w_{Rik} g_{Ri}(x_{Rk}) - w_{Iik} g_{Ri}(x_{Ik})) + \\ &+ j \sum_{k=1}^{N} (w_{Rik} g_{Ii}(x_{Ik}) + w_{Iik} g_{Ri}(x_{Rk})) \end{aligned} \tag{C.70}$$

or in a more compact form as:

$$\begin{aligned} &\begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_I \end{bmatrix} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \mathbf{r}_R \\ \mathbf{r}_I \end{bmatrix} \Rightarrow \tilde{\mathbf{u}} = \tilde{\mathbf{W}}\tilde{\mathbf{r}} = \tilde{\mathbf{W}}\mathbf{G}[\tilde{\mathbf{x}}] \\ &\tilde{\mathbf{z}} = \varphi[\tilde{\mathbf{y}}] \end{aligned} \tag{C.71}$$

in which $\mathbf{r}_R = [r_{R1}, \ldots, r_{RN}]^T = [g_{R1}(x_{R1}), \ldots, g_{RN}(x_{RN})]^T$ and similarly for the imaginary counterpart $\mathbf{r}_I$ and $\mathbf{h}(\tilde{\mathbf{u}}) = [h_1^R(u_{R1}), \ldots, h_N^R(y_{RN}), \ldots, h_1^I(y_{I1}), \ldots, h_N^I(y_{IN})]^T$ is the vector of activation functions. Then considering (C.71), the Jacobian matrix is defined as:

$$\det\left(\tilde{\mathbf{J}}\right) = \det \begin{bmatrix} \frac{\partial y_{R1}}{\partial x_{R1}} & \cdots & \frac{\partial y_{R1}}{\partial x_{I1}} & \cdots & \frac{\partial y_{R1}}{\partial x_{IN}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial y_{I1}}{\partial x_{R1}} & \cdots & \frac{\partial y_{I1}}{\partial x_{I1}} & \cdots & \frac{\partial y_{I1}}{\partial x_{IN}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial y_{IN}}{\partial x_{R1}} & \cdots & \frac{\partial y_{IN}}{\partial x_{I1}} & \cdots & \frac{\partial y_{IN}}{\partial x_{IN}} \end{bmatrix} \tag{C.72}$$

Each element of the Jacobian matrix can be rewritten as follows:

$$\begin{aligned} &\frac{\partial y_{Ri}}{\partial x_{Rk}} = \frac{\partial y_{Ri}}{\partial u_{Ri}} \left(\frac{\partial u_{Ri}}{\partial r_{Ri}} \frac{\partial r_{Ri}}{\partial x_{Rk}} + \frac{\partial u_{Ri}}{\partial r_{Ii}} \frac{\partial r_{Ii}}{\partial x_{Rk}} \right) \triangleq \dot{y}_{Ri} w_{Rik} \dot{r}_{Rk} \\ &\frac{\partial y_{Ri}}{\partial x_{Ik}} \triangleq -\dot{y}_{Ri} w_{Iik} \dot{r}_{Ik} \\ &\frac{\partial y_{Ii}}{\partial x_{Rk}} \triangleq \dot{y}_{Ii} w_{Iik} \dot{r}_{Rk} \\ &\frac{\partial y_{Ii}}{\partial x_{Ik}} \triangleq \dot{y}_{Ii} w_{Iik} \dot{r}_{Ik} \end{aligned} \tag{C.73}$$

Substitution of (C.73) into (C.72) yields

$$\det\left(\tilde{\mathbf{J}}\right) = \det\begin{bmatrix} \dot{y}_{R1}w_{R11}\dot{r}_{R1} & \cdots & -\dot{y}_{R1}w_{I11}\dot{r}_{I1} & \cdots & -\dot{y}_{R1}w_{I1N}\dot{r}_{IN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \dot{y}_{I1}w_{I11}\dot{r}_{R1} & \cdots & \dot{y}_{I1}w_{R11}\dot{r}_{I1} & \cdots & \dot{y}_{I1}w_{R1N}\dot{r}_{IN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \dot{y}_{IN}w_{IN1}\dot{r}_{R1} & \cdots & \dot{y}_{IN}w_{RN1}\dot{r}_{I1} & \cdots & \dot{y}_{IN}w_{RNN}\dot{r}_{IN} \end{bmatrix} =$$

$$\det\begin{bmatrix} \dot{y}_{R1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \dot{y}_{I1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \dot{y}_{IN} \end{bmatrix} \det\begin{vmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{vmatrix} \det\begin{bmatrix} \dot{r}_{R1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \dot{r}_{I1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \dot{r}_{IN} \end{bmatrix} \tag{C.74}$$

Then it follows

$$\det\left(\tilde{\mathbf{J}}\right) = \prod_{k=1}^{N} \dot{y}_{Rk}\dot{y}_{Ik}\dot{r}_{Rk}\dot{r}_{Ik} \det\left(\tilde{\mathbf{W}}\right) \tag{C.75}$$

Considering (C.75) in (C.68), neglecting the expectation and the pdf $p_{\mathbf{x}}(\mathbf{x})$ (which does not depend on learning parameters), the objective function becomes:

$$\max_{\mathbf{\Phi}}\left[\mathcal{L}\right] = \max_{\mathbf{\Phi}}\left[\begin{array}{l} \sum_{k=1}^{N} \ln \dot{y}_{Rk} + \sum_{k=1}^{N} \ln \dot{y}_{Ik} + \\ + \sum_{k=1}^{N} \ln \dot{r}_{Rk} + \sum_{k=1}^{N} \ln \dot{r}_{Ik} + \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) \end{array}\right] \tag{C.76}$$

The learning rule for the complex matrix $\mathbf{W}$ is composed by two expressions, for the real part and for the imaginary part: $\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I$. A simpler derivation is possible by considering the $2N \times 2N$ real matrix $\tilde{\mathbf{W}}$ so:

$$\Delta\tilde{\mathbf{W}} = \frac{\partial\mathcal{L}\left(\tilde{\mathbf{y}}, \mathbf{\Phi}\right)}{\partial\tilde{\mathbf{W}}} = \sum_{k=1}^{N} \frac{\partial}{\partial\tilde{\mathbf{W}}} \ln \dot{y}_{Rk} + \sum_{k=1}^{N} \frac{\partial}{\partial\tilde{\mathbf{W}}} \ln \dot{y}_{Ik} + \frac{\partial}{\partial\tilde{\mathbf{W}}} \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) \tag{C.77}$$

The gradient of the first two terms with respect to $\mathbf{W}_R$ and $\mathbf{W}_I$ gives

$$\begin{array}{ll} \sum_{k=1}^{N} \frac{\partial \ln \dot{y}_{Rk}}{\partial \mathbf{W}_R} = \mathbf{\Psi}_R \mathbf{r}_R^T; & \sum_{k=1}^{N} \frac{\partial \ln \dot{y}_{Ik}}{\partial \mathbf{W}_R} = \mathbf{\Psi}_R \mathbf{r}_I^T \\ \sum_{k=1}^{N} \frac{\partial \ln \dot{y}_{Rk}}{\partial \mathbf{W}_I} = -\mathbf{\Psi}_R \mathbf{r}_I^T; & \sum_{k=1}^{N} \frac{\partial \ln \dot{y}_{Ik}}{\partial \mathbf{W}_I} = \mathbf{\Psi}_I \mathbf{r}_R^T \end{array} \tag{C.78}$$

in which $\mathbf{\Psi}_R = \begin{bmatrix} \ddot{y}_{R1}/\dot{y}_{R1} \\ \vdots \\ \ddot{y}_{RN}/\dot{y}_{RN} \end{bmatrix}$ and $\mathbf{\Psi}_I = \begin{bmatrix} \ddot{y}_{I1}/\dot{y}_{I1} \\ \vdots \\ \ddot{y}_{IN}/\dot{y}_{IN} \end{bmatrix}$. From the last part of (C.77) results:

$$\frac{\partial}{\partial \tilde{\mathbf{W}}} \ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) = \left(\tilde{\mathbf{W}}^T\right)^{-1} \tag{C.79}$$

Considering that $\tilde{\mathbf{W}} \triangleq \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix}$, it is easy to show that:

$$\left(\tilde{\mathbf{W}}^T\right)^{-1} = \begin{bmatrix} \mathbf{W}_R^T & \mathbf{W}_I^T \\ -\mathbf{W}_I^T & \mathbf{W}_R^T \end{bmatrix}^{-1} = \begin{bmatrix} \mathrm{Re}\left\{\mathbf{W}^{-H}\right\} & -\mathrm{Im}\left\{\mathbf{W}^{-H}\right\} \\ \mathrm{Im}\left\{\mathbf{W}^{-H}\right\} & \mathrm{Re}\left\{\mathbf{W}^{-H}\right\} \end{bmatrix}$$

Starting from the learning rules (C.78) and (C.79) for the real matrix $\tilde{\mathbf{W}}$ it is possible to obtain the learning rule for the complex matrix $\mathbf{W}$:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{r}^H \tag{C.80}$$

where $\mathbf{\Psi} = \mathbf{\Psi}_R + j\mathbf{\Psi}_I$.
The learning rule (C.80) is similar to the one obtained for signals in real environment. The only difference is the substitution of the $(\bullet)^T$ transposition operator with the Hermitian operator $(\bullet)^H$.

In the same way it is possible to derive the learning rules for the control points $\mathbf{Q}_{\mathrm{G}}^{NL} = \left\{\mathbf{Q}_{\mathrm{Re(G)}}^{NL}, \mathbf{Q}_{\mathrm{Im(G)}}^{NL}\right\}$. Derivation of the cost function (C.76) with respect to the parameters yields:

$$\begin{aligned} \Delta\mathbf{Q}_{\mathrm{Re(G)},l,i}^{NL} &= \frac{\partial\mathcal{L}(\tilde{\mathbf{y}},\mathbf{\Phi})}{\partial\mathbf{Q}_{\mathrm{Re(G)},l,i}^{NL}} = \frac{\partial}{\partial\mathbf{Q}_{\mathrm{Re(G)},l,i}^{NL}}\left[\sum_{k=1}^{N}\left(\ln\dot{y}_{Rk} + \ln\dot{y}_{Ik}\right) + \ln\dot{r}_{Rl}\right] \\ \Delta\mathbf{Q}_{\mathrm{Im(G)},l,i}^{NL} &= \frac{\partial\mathcal{L}(\tilde{\mathbf{y}},\mathbf{\Phi})}{\partial\mathbf{Q}_{\mathrm{Im(G)},l,i}^{NL}} = \frac{\partial}{\partial\mathbf{Q}_{\mathrm{Im(G)},l,i}^{NL}}\left[\sum_{k=1}^{N}\left(\ln\dot{y}_{Rk} + \ln\dot{y}_{Ik}\right) + \ln\dot{r}_{Il}\right] \end{aligned} \tag{C.81}$$

in which $h = 1 \ldots N$ is the number of the spline neurons, i is the index of the interval of the spline function involved in the learning and $m = 1, \ldots, 4$ is the number of the control points inside the selected interval. Moreover

$$\begin{aligned} \Delta\mathbf{Q}_{\mathrm{Re(G)},l,i+m}^{NL} &= \frac{\dot{\mathbf{T}}_l\left(u_{\mathrm{Re}(x_l)}\right)(\mathbf{M})_m}{\dot{\mathbf{T}}_l\left(u_{\mathrm{Re}(x_l)}\right)\mathbf{M}\mathbf{Q}_{\mathrm{Re(G)},l,i+m}} + \\ &+\mathrm{Re}\left[\mathbf{\Psi}\left(\mathbf{W}^H\right)_l\right]\tfrac{1}{2}\mathbf{T}_l\left(u_{\mathrm{Re}(x_l)}\right)(\mathbf{M})_m \\ \Delta\mathbf{Q}_{\mathrm{Im(G)},l,i+m}^{NL} &= \frac{\dot{\mathbf{T}}_l\left(u_{\mathrm{Im}(x_l)}\right)(\mathbf{M})_m}{\dot{\mathbf{T}}_l\left(u_{\mathrm{Im}(x_l)}\right)\mathbf{M}\mathbf{Q}_{\mathrm{Im(G)},l,i+m}} + \\ &+\mathrm{Im}\left[\mathbf{\Psi}\left(\mathbf{W}^H\right)_l\right]\tfrac{1}{2}\mathbf{T}_l\left(u_{\mathrm{Im}(x_l)}\right)(\mathbf{M})_m \end{aligned} \tag{C.82}$$

In (C.82) the expression $(\mathbf{M})_m$ is a vector composed with the m-th columns of the matrix $\mathbf{M}$.

In the same manner derivation with respect to the parameters $\mathbf{Q}_h^{SC} = \left\{\mathbf{Q}_{h^R}^{SC}, \mathbf{Q}_{h^I}^{SC}\right\}$ yields

$$\begin{aligned}\Delta\mathbf{Q}_{h^R,l,i}^{SC} &= \frac{\partial\mathcal{L}(\tilde{\mathbf{y}},\mathbf{\Phi})}{\partial\mathbf{Q}_{h^R,l,i}^{SC}} = \frac{\partial}{\partial\mathbf{Q}_{h^R,l,i}^{NL}}\sum_{k=1}^{N}\ln\dot{y}_{Rk} \\ \Delta\mathbf{Q}_{h^I,l,i}^{SC} &= \frac{\partial\mathcal{L}(\tilde{\mathbf{y}},\mathbf{\Phi})}{\partial\mathbf{Q}_{h^I,l,i}^{SC}} = \frac{\partial}{\partial\mathbf{Q}_{h^I,l,i}^{NL}}\sum_{k=1}^{N}\ln\dot{y}_{Ik}\end{aligned} \tag{C.83}$$

and then

$$\begin{aligned}\Delta\mathbf{Q}_{h^R,l,i+m}^{SC} &= \frac{\dot{\mathbf{T}}_l\left(u_{\mathrm{Re}(y_l)}\right)(\mathbf{M})_m}{\dot{\mathbf{T}}_l\left(u_{\mathrm{Re}(y_l)}\right)\mathbf{M}\mathbf{Q}_{h^R,l,i+m}} \\ \Delta\mathbf{Q}_{h^I,l,i+m}^{SC} &= \frac{\dot{\mathbf{T}}_l\left(u_{\mathrm{Im}(y_l)}\right)(\mathbf{M})_m}{\dot{\mathbf{T}}_l\left(u_{\mathrm{Im}(y_l)}\right)\mathbf{M}\mathbf{Q}_{h^I,l,i+m}}\end{aligned} \tag{C.84}$$

C.4 The Nonlinear Case: MMI Algorithm

This section presents the derivation of the learning rule for the complex algorithm in the case of application of Minimization of Mutual Information (MMI) algorithm.

Given the model in the following Figure C.2 using the splitting function (5.3) we can write:

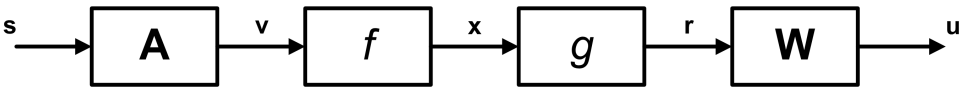

Fig. C.2: PNL mixing-demixing model

$$\mathbf{x}[n] = \mathbf{f}_R(v_R[n]) + j\mathbf{f}_I(v_I[n]) \tag{C.85}$$

where $\mathbf{v}[n] = \mathbf{v}_R[n] + j\mathbf{v}_I[n] = \mathbf{A}\mathbf{s} = (\mathbf{A}_R + j\mathbf{A}_I)(\mathbf{s}_R + j\mathbf{s}_I)$.
For the de-mixing model we have:

$$\mathbf{r}[n] = \mathbf{g}_R(\mathbf{f}_R(\mathbf{v}_R)) + j\mathbf{g}_I(\mathbf{f}_I(\mathbf{v}_I)) \tag{C.86}$$

$$\mathbf{u}[n] = \mathbf{u}_R[n] + j\mathbf{u}_I[n] = \mathbf{W}\mathbf{r} = (\mathbf{W}_R + j\mathbf{W}_I)(\mathbf{r}_R + j\mathbf{r}_I) \tag{C.87}$$

or esplicitely

$$\begin{bmatrix}\mathbf{u}_R \\ \mathbf{u}_I\end{bmatrix} = \begin{bmatrix}\mathbf{w}_R & -\mathbf{w}_I \\ \mathbf{w}_I & \mathbf{w}_R\end{bmatrix}\begin{bmatrix}\mathbf{r}_R \\ \mathbf{r}_I\end{bmatrix}$$

So the n-th output is

$$u_n[n] = \sum_{k=1}^{N}(w_{Rnk}r_{Rk}[n] - w_{Ink}r_{Ik}[n]) + j\sum_{k=1}^{N}(w_{Ink}r_{Rk}[n] + w_{Rnk}r_{Ik}[n]) \tag{C.88}$$

Now the Mutual Information [55] of the network output is

$$\begin{aligned}\mathbf{I}(\tilde{\mathbf{u}}) = \mathbf{I}(\mathbf{u}_R, \mathbf{u}_I) = \int p_{\mathbf{u}}(\mathbf{u}) \frac{\log p_{\mathbf{u}}(\mathbf{u})}{\prod_{i=1}^{N} p_{u_i}(u_i)} d\mathbf{u} = \\ = E\left\{\log(p_{\mathbf{u}}(\mathbf{u}))\right\} - \sum_{i=1}^{N} E\left\{\log p_{u_i}(u_i)\right\}\end{aligned} \tag{C.89}$$

under the assumption that the real and imaginary random variable are independent:

$$p_{u_i}(u_i) = p_{u_{Ri}}(u_{Ri}) \cdot p_{u_{Ii}}(u_{Ii})$$

Let we pose

$$\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_I \end{bmatrix} \tag{C.90}$$

so the contrast function $\mathcal{L}$ can be written as

$$\begin{aligned}\mathcal{L} = \mathbf{I}(\tilde{\mathbf{u}}) = E\left\{\log p_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}})\right\} - \sum_{i=1}^{N} E\left\{\log p_{u_{Ri}}(u_{Ri})\right\} + \\ - \sum_{i=1}^{N} E\left\{\log p_{u_{Ii}}(u_{Ii})\right\}\end{aligned} \tag{C.91}$$

Because is [144]

$$p_{\tilde{\mathbf{u}}}(\tilde{\mathbf{u}}) = \frac{p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})}{\det\left|\tilde{\mathbf{J}}\right|} \tag{C.92}$$

from (C.92) we obtain

$$\begin{aligned}\mathcal{L} = E\left\{\log p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})\right\} - E\left\{\log\left|\tilde{\mathbf{J}}\right|\right\} + \\ - \sum_{i=1}^{N} E\left\{\log p_{u_{Ri}}(u_{Ri})\right\} - \sum_{i=1}^{N} E\left\{\log p_{u_{Ii}}(u_{Ii})\right\}\end{aligned} \tag{C.93}$$

The Jacobian matrix $\tilde{\mathbf{J}}$ can be expressed as

$$\det \mathbf{J} = \begin{bmatrix} \frac{\partial u_{R1}}{\partial x_{R1}} & \dots & \frac{\partial u_{R1}}{\partial x_{RN}} & \frac{\partial u_{R1}}{\partial x_{I1}} & \dots & \frac{\partial u_{R1}}{\partial x_{IN}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_{RN}}{\partial x_{R1}} & \dots & \frac{\partial u_{RN}}{\partial x_{RN}} & \frac{\partial u_{RN}}{\partial x_{I1}} & \dots & \frac{\partial u_{RN}}{\partial x_{IN}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_{IN}}{\partial x_{R1}} & \dots & \frac{\partial u_{IN}}{\partial x_{RN}} & \frac{\partial u_{IN}}{\partial x_{I1}} & \dots & \frac{\partial u_{IN}}{\partial x_{IN}} \end{bmatrix} \tag{C.94}$$

The single terms can be obtained as

$$\begin{aligned}\frac{\partial u_{Rk}}{\partial x_{Rn}} &= \frac{\partial u_{Rk}}{\partial r_{Rn}} \cdot \frac{\partial r_{Rn}}{\partial x_{Rn}} = \dot{g}_{Rn}(x_{Rn}) w_{Rkn} \\ \frac{\partial u_{Ik}}{\partial x_{Rn}} &= \frac{\partial u_{Ik}}{\partial r_{Rn}} \cdot \frac{\partial r_{Rn}}{\partial x_{Rn}} = \dot{g}_{Rn}(x_{Rn}) w_{Ikn} \\ \frac{\partial u_{Rk}}{\partial x_{In}} &= \frac{\partial u_{Rk}}{\partial r_{In}} \cdot \frac{\partial r_{In}}{\partial x_{In}} = -\dot{g}_{In}(x_{In}) w_{Ikn} \\ \frac{\partial u_{Ik}}{\partial x_{In}} &= \frac{\partial u_{Ik}}{\partial r_{In}} \cdot \frac{\partial r_{In}}{\partial x_{In}} = \dot{g}_{In}(x_{In}) w_{Rkn}\end{aligned}$$

so the expression of the Jacobian (C.94) becomes

$$\begin{aligned}
\det \mathbf{J} &= \begin{bmatrix} w_{R11}\dot{g}_{R1} & \cdots & w_{R1N}\dot{g}_{RN} & -w_{I11}\dot{g}_{I1} & \cdots & -w_{R11}\dot{g}_{IN} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{RN1}\dot{g}_{R1} & \cdots & w_{RNN}\dot{g}_{RN} & -w_{IN1}\dot{g}_{I1} & \cdots & -w_{INN}\dot{g}_{IN} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{IN1}\dot{g}_{R1} & \cdots & w_{INN}\dot{g}_{RN} & w_{RN1}\dot{g}_{I1} & \cdots & w_{RNN}\dot{g}_{IN} \end{bmatrix} = \\
&= \det \left[\begin{pmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{pmatrix} \begin{pmatrix} \dot{g}_{R1} & & & & & \\ & \ddots & & & & \\ & & \dot{g}_{RN} & & & \\ & & & \dot{g}_{I1} & & \\ & & & & \ddots & \\ & & & & & \dot{g}_{IN} \end{pmatrix} \right] = \\
&= \det(\tilde{\mathbf{W}}) \prod_{i=1}^{N} \dot{g}_{Ri} \prod_{i=1}^{N} \dot{g}_{Ii}
\end{aligned} \tag{C.95}$$

Let we pose $\mathbf{\Phi} = \left\{ w_{i,j}, Q_G^{NL} \right\}$ the free network parameter, we ha the following learning rule:

$$\begin{aligned}
\tfrac{\partial \mathcal{L}}{\partial \mathbf{\Phi}} &= -\tfrac{\partial}{\partial \mathbf{\Phi}} \log \left[\det \left(\tilde{\mathbf{W}} \right) \right] + \\
&- \left(\tfrac{\partial}{\partial \mathbf{\Phi}} \sum_{i=1}^{N} \log \dot{g}_{Ri}(x_{Ri}) - \tfrac{\partial}{\partial \mathbf{\Phi}} \sum_{i=1}^{N} \log \dot{g}_{Ii}(x_{Ii}) \right) + \\
&- \sum_{i=1}^{N} \left(\Psi_{Ri}(u_{Ri}) \tfrac{\partial u_{Ri}}{\partial \Phi} + \Psi_{Ii}(u_{Ii}) \tfrac{\partial u_{Ii}}{\partial \Phi} \right) = A + B + C
\end{aligned} \tag{C.96}$$

where A, B and C are the three terms of the previous equation (C.96).

For the learning rule of the $\mathbf{W}$ we need the terms A and C:

$$\begin{aligned}
\tfrac{\partial C}{\partial W_{Rn,k}} &= \Psi_{Rn}(u_{Rn}) r_{Rk} + \Psi_{In}(u_{In}) r_{Ik} \\
\tfrac{\partial C}{\partial W_{In,k}} &= -\Psi_{Rn}(u_{Rn}) r_{Ik} + \Psi_{In}(u_{In}) r_{Rk}
\end{aligned} \tag{C.97}$$

Using (C.97) we obtain:

$$\begin{aligned}
&\Delta \mathbf{W} = \Delta \mathbf{W}_R + j \Delta \mathbf{W}_I = \\
&= \sum_{i=1}^{N} \left(\Psi_{Ri} \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{r}_R^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} + \Psi_{Ii} \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{r}_I^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \right) + j \sum_{i=1}^{N} \left(\Psi_{Ii} \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{r}_R^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} - \Psi_{Ri} \begin{bmatrix} \mathbf{0}^T \\ \vdots \\ \mathbf{r}_I^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \right) = \\
&= \left(\mathbf{\Psi}_R + j \mathbf{\Psi}_I \right) \left(\mathbf{r}_R - j \mathbf{r}_I \right) = \mathbf{\Psi} \mathbf{r}^H
\end{aligned} \tag{C.98}$$

Similarly for the term A we have $\frac{\partial A}{\partial \tilde{\mathbf{W}}} = \tilde{\mathbf{W}}^{-T}$, from which we derive:

$$\frac{\partial A}{\partial \mathbf{W}} = \mathbf{W}^{-H} \tag{C.99}$$

Using the equation (C.98) and (C.99) we obtain the learning rule for the adaptation of the $\mathbf{W}$ matrix:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = -\left(\mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{r}^{H}\right) \tag{C.100}$$

For the learning rule of the $\mathbf{Q}_G^{NL}$ we need the term B:

$$\frac{\partial B}{\partial Q^{NL}_{G_R,n,m}} = -\frac{1}{\dot{g}_{Rn}} \cdot \frac{\partial}{\partial Q^{NL}_{G_R,n,m}} \cdot \dot{g}_{Rn} - \sum_{i=1}^{N} \Psi_{Ri} \frac{\partial u_{Ri}}{\partial Q^{NL}_{G_R,n,m}} - \sum_{i=1}^{N} \Psi_{Ii} \frac{\partial u_{Ii}}{\partial Q^{NL}_{G_R,n,m}} \tag{C.101}$$

and remembering that

$$\begin{array}{l} \frac{\partial u_{Ri}}{\partial Q^{NL}_{G_R,n,m}} = \frac{\partial u_{Ri}}{\partial r_{Rn}} \cdot \frac{\partial r_{Rn}}{\partial Q^{NL}_{G_R,n,m}} = w_{Rin}\mathbf{T}_{Rn}\mathbf{M}_m \\ \frac{\partial u_{Ii}}{\partial Q^{NL}_{G_R,n,m}} = \frac{\partial u_{Ii}}{\partial r_{Rn}} \cdot \frac{\partial r_{Rn}}{\partial Q^{NL}_{G_R,n,m}} = w_{Iin}\mathbf{T}_{Rn}\mathbf{M}_m \end{array} \tag{C.102}$$

Using (C.101) and (C.102) we obtain the learning rule for the real part of the Q^{g_R}:

$$\frac{\partial B}{\partial Q^{NL}_{G_R,n,m}} = -\frac{\dot{\mathbf{T}}_{Rn}\mathbf{M}_m}{\dot{\mathbf{T}}_{Rn}\mathbf{M}\mathbf{Q}^{NL}_{G_R,n,m}} - \sum_{i=1}^{N} (\Psi_{Ri}w_{Rin} + \Psi_{Ii}w_{Iin})\mathbf{T}_{Rn}\mathbf{M}_m \tag{C.103}$$

And similarly for the imaginary part:

$$\frac{\partial B}{\partial Q^{NL}_{G_I,n,m}} = -\frac{1}{\dot{g}_{In}} \cdot \frac{\partial}{\partial Q^{NL}_{G_I,n,m}} \cdot \dot{g}_{In} - \sum_{i=1}^{N} \Psi_{Ri} \frac{\partial u_{Ri}}{\partial Q^{NL}_{G_I,n,m}} - \sum_{i=1}^{N} \Psi_{Ii} \frac{\partial u_{Ii}}{\partial G_I, n, m^{NL}} \tag{C.104}$$

and remembering that

$$\begin{array}{l} \frac{\partial u_{Ri}}{\partial Q^{NL}_{G_I,n,m}} = \frac{\partial u_{Ri}}{\partial r_{In}} \cdot \frac{\partial r_{In}}{\partial Q^{NL}_{G_I,n,m}} = -w_{Iin}\mathbf{T}_{In}\mathbf{M}_m \\ \frac{\partial u_{Ii}}{\partial Q^{NL}_{G_I,n,m}} = \frac{\partial u_{Ii}}{\partial r_{In}} \cdot \frac{\partial r_{In}}{\partial Q^{NL}_{G_I,n,m}} = w_{Rin}\mathbf{T}_{In}\mathbf{M}_m \end{array} \tag{C.105}$$

Using (C.104) and (C.105) we obtain the learning rule for the real part of the Q^{g_I}:

$$\frac{\partial B}{\partial Q^{NL}_{G_I,n,m}} = -\frac{\dot{\mathbf{T}}_{In}\mathbf{M}_m}{\dot{\mathbf{T}}_{In}\mathbf{M}\mathbf{Q}^{NL}_{G_I,n,m}} - \sum_{i=1}^{N} (-\Psi_{Ri}w_{Iin} + \Psi_{Ii}w_{Rin})\mathbf{T}_{In}\mathbf{M}_m \tag{C.106}$$

The learning rules for the score functions are obtained in a direct way [208] minimizing the following mean square error:

$$\varepsilon_j = \frac{1}{2} E\left\{ \left| \tilde{\Psi}_j - \frac{\dot{p}_{\tilde{u}_j}(\tilde{\mathbf{u}})}{p_{u_j}(\tilde{\mathbf{u}})} \right|^2 \right\} \tag{C.107}$$

where $\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_I \end{bmatrix}$, $\mathbf{\Psi}_R = \frac{\dot{p}_{\mathbf{u}R}(\mathbf{u}_R)}{p_{\mathbf{u}R}(\mathbf{u}_R)}$ and $\mathbf{\Psi}_I = \frac{\dot{p}_{\mathbf{u}I}(\mathbf{u}_I)}{p_{\mathbf{u}\mathbf{I}}(\mathbf{u}_I)}$. In order to obtain the learning rules, we do the derivative of ε_j with respect to the generic vector Q^{SC}:

$$\begin{array}{c} \frac{\partial \varepsilon_j}{\partial Q^{SC}} = E\left\{ \left(\tilde{\Psi}_j - \frac{p'_{u_j}(u_j)}{p_{u_j}(u_j)} \right) \cdot \frac{\partial \tilde{\Psi}_j}{\partial Q^{SC}} \right\} = \\ \simeq \tilde{\Psi}_j \frac{\partial \tilde{\Psi}_j}{\partial Q^{SC}} + \frac{\partial^2 \tilde{\Psi}_j}{\partial u_j \partial Q^{SC}} \end{array} \tag{C.108}$$

where we have approximated the expectation value and we have used the Lemma (8.5).

Finally using (C.108) we obtain the learning rules:

$$\begin{array}{l} \frac{\partial \varepsilon}{\partial Q_R^{SC}} = \mathbf{T_R M T_R M Q}_R^{SC} - \frac{1}{\Delta} \dot{\mathbf{T}}_R \mathbf{M} \\ \frac{\partial \varepsilon}{\partial Q_I^{SC}} = \mathbf{T_I M T_I M Q}_I^{SC} - \frac{1}{\Delta} \dot{\mathbf{T}}_I \mathbf{M} \end{array} \tag{C.109}$$

C.5 Algorithm with Renyi's Entropy

An analogue algorithm can be obtained considering the Renyi's Entropy which is denoted by H_{R_α}:

$$H_{R_\alpha} = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{+\infty} \left(p_{\mathbf{y}}(\mathbf{y}) \right)^\alpha d\mathbf{y} \right) \tag{C.110}$$

In fact applying the eq. (C.110) to the joint network output $p_{\mathbf{y}}(\mathbf{y})$ and remembering eq. (C.7), we obtain:

$$H_{R_\alpha} = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{+\infty} \left(\frac{p_{\mathbf{x}}(\mathbf{x})}{|\mathbf{J}|} \right)^\alpha d\mathbf{x} \right) \tag{C.111}$$

Now the Jacobian matrix $\mathbf{J}$ can be expressed as follows:

$$
\begin{aligned}
|\mathbf{J}| &= \begin{vmatrix} w_{11}g'_1 & w_{12}g'_1 & \cdots & w_{1N}g'_1 \\ w_{21}g'_2 & w_{22}g'_2 & \cdots & w_{2N}g'_2 \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1}g'_N & w_{N2}g'_N & \cdots & w_{NN}g'_N \end{vmatrix} = \\
&= \begin{vmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{vmatrix} \cdot \begin{vmatrix} g'_1 & 0 & \cdots & 0 \\ 0 & g'_2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g'_N \end{vmatrix} = \\
&= |\mathbf{W}| \cdot \prod_{i=1}^{N} g'_i
\end{aligned}
\tag{C.112}
$$

Introducing eq. (C.112) in eq. (C.111) it is possible to write:

$$
\begin{aligned}
H_{R_\alpha} &= \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{+\infty} \left(\frac{p_{\mathbf{x}}(\mathbf{x})}{|\mathbf{W}| \cdot \prod_{i=1}^{N} g'_i} \right)^{\alpha} d\mathbf{x} \right) = \\
&= \frac{\log \left(\int_{-\infty}^{+\infty} p_{\mathbf{x}}^{\alpha}(\mathbf{x}) d\mathbf{x} \right) - \alpha \log |\mathbf{W}| - \alpha \sum_{i=1}^{N} \log g'_i}{1-\alpha}
\end{aligned}
\tag{C.113}
$$

Since the first term does not depend by the matrix weights w_{ij}, the learning rule for the $\mathbf{W}$ matrix becomes:

$$
\begin{aligned}
\Delta w_{ij} \propto \frac{\partial H_{R_\alpha}}{\partial w_{ij}} &= -\frac{\alpha}{1-\alpha} \left[\frac{\partial}{\partial w_{ij}} \log |\mathbf{W}| + \frac{\partial}{\partial w_{ij}} \left(\sum_{i=1}^{N} \log g'_i \right) \right] = \\
&= -\frac{\alpha}{1-\alpha} \left[\frac{\partial}{\partial w_{ij}} \log |\mathbf{W}| + \sum_{i=1}^{N} \frac{g''_i}{g'_i} x_j \right]
\end{aligned}
\tag{C.114}
$$

The eq. (C.114) can be rewritten in matrix notation as:

$$
\Delta \mathbf{W} \propto \frac{\partial H_{R_\alpha}}{\partial \mathbf{W}} = -\frac{\alpha}{1-\alpha} \left[\mathbf{W}^{-H} + \mathbf{\Psi} \mathbf{x}^{H} \right] \tag{C.115}
$$

We can see that eq. (C.115) is formally identical to eq. (C.39): the only thing that changes is a constant term which can be adsorbed in the learning rate.

D

Mathematical Background

—If knowledge can create problems,
it is not through ignorance that we can solve them.
I. Asimov

THE aims of this appendix is to introduce some quite obscure mathematical concepts, which are sometimes not much known from usual readers, such as the Kronecker product. See [117] for more details.

D.1 Vectorization operator

In mathematics, especially in linear algebra and matrix theory, the *vectorization* of a matrix is a linear transformation which converts the matrix into a *column vector*. Specifically, the vectorization of an $m \times n$ matrix A, denoted by $\mathrm{vec}(A)$, is the $mn \times 1$ column vector obtain by stacking the columns of the matrix A on top of one another:

$$\mathrm{vec}(A) = [a_{11}, \ldots, a_{m1}, a_{12}, \ldots, a_{m2}, \ldots, a_{1n}, \ldots, a_{mn}]^T \tag{D.1}$$

Here a_{ij} represents the (i, j)-th element of matrix A and the superscript T denotes the transpose operator. Let we give an easy example.

Example 13.

For the following 2×2 matrix $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, the vectorization is

$$\mathrm{vec}\,(A) = \begin{bmatrix} 1 \\ 3 \\ \hline 2 \\ 4 \end{bmatrix}$$

The vec operator is very useful in computational applications.

D.2 Kronecker product

In mathematics, the Kronecker product, denoted by $\otimes$, is an operation on two matrices of arbitrary size resulting in a block matrix. It is a special case of a *tensor product*. The Kronecker product should not be confused with the usual matrix multiplication, which is an entirely different operation. It is named after German mathematician Leopold Kronecker.

If A is an $m \times n$ matrix and B is a $p \times q$ matrix, then the *Kronecker product* $A \otimes B$ is the $mp \times nq$ block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{D.2}$$

More explicitly, we have

$$A \otimes B = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix}$$

In other words, each entry of A is replaced by a scaled multiple of B. We show a simple example of how do the Kronecker product.

Example 14.
It is calculated the Kronecker product of the following two matrices:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 0 & 1 \cdot 5 & 2 \cdot 0 & 2 \cdot 5 \\ 1 \cdot 6 & 1 \cdot 7 & 2 \cdot 6 & 2 \cdot 7 \\ 3 \cdot 0 & 3 \cdot 5 & 4 \cdot 0 & 4 \cdot 5 \\ 3 \cdot 6 & 3 \cdot 7 & 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

In particular, if A is the identity matrix I_m of order m, then $I_m \otimes B$ is a block diagonal matrix with B repeated along its diagonal:

$$I_m \otimes B = \text{diag}\left(\underbrace{B, \quad B, \quad \ldots, \quad B}_{m \text{ times}} \right)$$

On the other hand, $A \otimes I_n$ is not a block diagonal matrix, as shown by the following

Example 15.

If $m = n = 2$, $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ and $B = I_2$, then we obtain

$$A \otimes I_2 = \left[\begin{array}{cc|cc} a_{11} & 0 & a_{12} & 0 \\ 0 & a_{11} & 0 & a_{12} \\ \hline a_{21} & 0 & a_{22} & 0 \\ 0 & a_{21} & 0 & a_{22} \end{array}\right]$$

One of the main uses of Kronecker products is that they allows us to replace matrix operations by vector operations.

The following Lemma is a non-comprehensive list of useful properties of Kronecker products.

Lemma 19 *Consider $m \times m$ and $n \times n$ matrices A and B and let $\{\alpha_i, i = 1, \ldots, m\}$ and $\{\beta_j, j = 1, \ldots, n\}$ denote their eigenvalues, respectively. The matrices be real or complex-valued. Then it holds that:*

1. *The Kronecker product is a special case of the tensor product, so it is bilinear and associative*

$$\begin{aligned} &A \otimes (B + C) = A \otimes B + A \otimes C, \\ &(A + B) \otimes C = A \otimes C + B \otimes C, \\ &(kA) \otimes B = A \otimes (kB) = k\,(A \otimes B)\,, \\ &(A \otimes B) \otimes C = A \otimes (B \otimes C)\,, \end{aligned}$$

   where C is an opportune matrix and k a scalar.

2. *The Kronecker product is not commutative: in general, $A \otimes B$ and $B \otimes A$ are different matrices.*

   - *However, $A \otimes B$ and $B \otimes A$ are permutation equivalent, meaning that there exist permutation matrices P and Q such that*

$$A \otimes B = P(B \otimes A)Q$$

- *If A and B are square matrices, then $A \otimes B$ and $B \otimes A$ are even permutation similar, meaning that we can take $P = Q^T$.*

3. *If C and D are matrices of such size that one can form the matrix products AC and BD, then*

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

4. *If A and B are invertible, then*

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

5. $(A \otimes B)^T = A^T \otimes B^T$ *as well as* $(A \otimes B)^* = A^* \otimes B^*$.

6. $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B)$.

7. $\mathrm{Rank}(A \otimes B) = \mathrm{Rank}(A)\mathrm{Rank}(B)$.

8. $\det(A \otimes B) = (\det A)^n (\det B)^m$.

9. $(A \otimes B)$ *has mn eigenvalues ad they are equal to all combinations $\{\alpha_i \beta_j\}$, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.*

10. *For any matrices A, B, C of compatible dimensions, it holds*

$$\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$$

There are two other useful formulations:

- $\mathrm{vec}(ABC) = (I \otimes AB)\mathrm{vec}(C) = (C^T B^T \otimes I)\mathrm{vec}(A)$;
- $\mathrm{vec}(AB) = (I \otimes A)\mathrm{vec}(B) = (B^T \otimes I)\mathrm{vec}(A)$.

In addition if A is $n \times n$, B is $m \times m$ and I_k denotes the $k \times k$ identity matrix then we can define the *Kronecker sum*, $\oplus$, by

$$A \oplus B = A \otimes I_m + I_n \otimes B. \tag{D.3}$$

We have the following formula for the *matrix exponential* which is useful in the numerical evaluation of certain continuous-time Markov processes

$$e^{A \oplus B} = e^A \otimes e^B.$$

The Kronecker product can be used to get a convenient representation for some matrix equations. Consider for instance the equation $AXB = C$, where A, B and C are given matrices and the matrix X is the unknown. We can rewrite this equation as

$$(B^T \otimes A)\mathrm{vec}(X) = \mathrm{vec}(AXB) = \mathrm{vec}(C)$$

It now follows from the properties of the Kronecker product that the equation $AXB = C$ has a unique solution if and only if A and B are nonsingular [84].

D.3 Tracy-Singh product

Let the $m \times n$ matrix A be partitioned into the $m_i \times n_j$ blocks A_{ij} and $p \times q$ matrix B into the $p_k \times q_l$ blocks B_{kl} with of course $\sum_i m_i = m$, $\sum_j n_j = n$, $\sum_k p_k = p$ and $\sum_l q_l = q$.

The *Tracy-Singh product* [84] is defined as

$$A \circ B = (A_{ij} \circ B)_{ij} = ((A_{ij} \otimes B_{kl})_{kl})_{ij} \tag{D.4}$$

which means that the (i, j)-th sub-block of the $mp \times nq$ product $A \circ B$ is the $m_i p \times n_j q$ matrix $A_{ij} \circ B$, of which the (k, l)-th sub-block equals the $m_i p_k \times n_j q_l$ matrix $A_{ij} \otimes B_{kl}$. Essentially the *Tracy-Singh product* is the pairwise Kronecker product for each pair of partitions in the two matrices. We will introduce these concepts with the following example.

Example 16.

If A and B both are 2×2 partitioned matrices, e.g.:

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ \hline 7 & 8 & 9 \end{array}\right], \quad B = \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right] = \left[\begin{array}{c|cc} 1 & 4 & 7 \\ \hline 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}\right]$$

we get:

$$A \circ B = \left[\begin{array}{c|c} A_{11} \circ B & A_{12} \circ B \\ \hline A_{21} \circ B & A_{22} \circ B \end{array}\right] = \left[\begin{array}{c|c|c|c} A_{11} \otimes B_{11} & A_{11} \otimes B_{12} & A_{12} \otimes B_{11} & A_{12} \otimes B_{12} \\ \hline A_{11} \otimes B_{21} & A_{11} \otimes B_{22} & A_{12} \otimes B_{21} & A_{12} \otimes B_{22} \\ \hline A_{21} \otimes B_{11} & A_{21} \otimes B_{12} & A_{22} \otimes B_{11} & A_{22} \otimes B_{12} \\ \hline A_{21} \otimes B_{21} & A_{21} \otimes B_{22} & A_{22} \otimes B_{21} & A_{22} \otimes B_{22} \end{array}\right] =$$

$$= \left[\begin{array}{cc|cc|cc|cc} 1 & 2 & 4 & 7 & 8 & 14 & 3 & 12 & 21 \\ 4 & 5 & 16 & 28 & 20 & 35 & 6 & 24 & 42 \\ \hline 2 & 4 & 5 & 8 & 10 & 16 & 6 & 15 & 24 \\ 3 & 6 & 6 & 9 & 12 & 18 & 9 & 18 & 27 \\ 8 & 10 & 20 & 32 & 25 & 40 & 12 & 30 & 48 \\ 12 & 15 & 24 & 36 & 30 & 45 & 18 & 36 & 54 \\ \hline 7 & 8 & 28 & 49 & 32 & 56 & 9 & 36 & 63 \\ \hline 14 & 16 & 35 & 56 & 40 & 64 & 18 & 45 & 72 \\ 21 & 24 & 42 & 63 & 48 & 72 & 27 & 54 & 81 \end{array}\right]$$

D.4 Khatri-Rao product

Let the $m \times n$ matrix A be partitioned into the $m_i \times n_j$ blocks A_{ij} and $p \times q$ matrix B into the $p_k \times q_l$ blocks B_{kl} with of course $\sum_i m_i = m$, $\sum_j n_j = n$, $\sum_k p_k = p$ and $\sum_l q_l = q$.

The *Khatri-Rao product* [84] is defined as

$$A * B = (A_{ij} \otimes B_{ij})_{ij} \tag{D.5}$$

in which the (i, j)-th block is the $m_i p_i \times n_j q_j$ sized Kronecker product of the corresponding blocks of A and B, assuming the number of row and column partitions of both matrices is equal. The size of the product is then $\sum_i m_i p_i \times \sum_j n_j q_j$. With an example:

Example 17.

Proceeding with the same matrices as the previous example we obtain:

$$A * B = \left[\begin{array}{c|c} A_{11} \otimes B_{11} & A_{12} \otimes B_{12} \\ \hline A_{21} \otimes B_{21} & A_{22} \otimes B_{22} \end{array}\right] = \left[\begin{array}{cc|cc} 1 & 2 & 12 & 21 \\ 4 & 5 & 24 & 42 \\ \hline 14 & 16 & 45 & 72 \\ 21 & 24 & 54 & 81 \end{array}\right]$$

A column-wise Kronecker product of two matrices may also be called the *Khatri-Rao product*. This product assumes the partitions of the matrices are their columns. In this case $m_1 = m, p_1 = p, n = q$ and $\forall j : n_j = p_j = 1$. The resulting product is a $mp \times n$ matrix of which each column is the Kronecker product of the corresponding columns of A and B. Let we explain that with an example.

Example 18.

Using the matrices from the previous examples with the columns partitioned:

$$C = \left[\begin{array}{c|c|c} C_1 & C_2 & C_3 \end{array}\right] = \left[\begin{array}{c|c|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}\right],$$
$$D = \left[\begin{array}{c|c|c} D_1 & D_2 & D_3 \end{array}\right] = \left[\begin{array}{c|c|c} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array}\right],$$

so that:

$$C * D = \left[\begin{array}{c|c|c} C_1 \otimes D_1 & C_2 \otimes D_2 & C_3 \otimes D_3 \end{array}\right] = \left[\begin{array}{c|c|c} 1 & 8 & 21 \\ 2 & 10 & 24 \\ 3 & 12 & 27 \\ 4 & 20 & 42 \\ 8 & 25 & 48 \\ 12 & 30 & 54 \\ 7 & 32 & 63 \\ 14 & 40 & 72 \\ 21 & 48 & 81 \end{array}\right].$$

Acknowledgements

—Thank you India
Thank you terror
Thank you disillusionment
Thank you frailty
Thank you consequence
Thank you, thank you silence.
A. Morissette

FIRST *of all I have to say a big thank you to my Ph.D. supervisor, Prof. Aurelio Uncini, which is a continuous and successful source of new inspiration. I have to confess that his interest in research is exceptional for me, very motivational.*

Secondly I have to say thank you to all my new colleagues which encourage me in new and new interesting problems. In particular my esteem goes, in random order, to Antonello Rizzi, Massimo Panella, Fabio Massimo Frattale Mascioli and Raffaele Parisi, which are not only friends and colleagues but above all "magistri vitae".

A great thanks goes to Daniele Vigliano which taught to me the particular topics of this thesis and gave me the opportunity to begin the work.

I cannot release to thank you my friends and colleagues in Ph.D. studentship, which support me in my doubts and give me lots of helpful suggestions. Thanks to Gabriele Bunkheila, Albenzio Cirillo, Mario Antonelli, Nicola Buccino and Timothy Battisti.

A very special thanks goes to my girlfriend Corinna, who has borne me in this difficult and very stressful time being close to me, and has found the sweet and exact words of encouragement: thanks a lot!

Thanks to my family who has supported me for all this long time, in my studies, and has allowed me to arrive so far.

Thanks a lot to all my friends, both visible and concrete, that invisible and surreal, which, with their existence, allowed me to be what I am like.

Biographical Sketch

MICHELE SCARPINITI was born in Leonberg, Germany, on February 3rd, 1978. At the age of 6 he returned to Torre Melissa, near Crotone, Italy, where he lived till the age of 19. Here he attended high school (Liceo Scientifico statale "FILOLAO"). Then he moved to Rome in 1997 to attend the School of Engineering (University of Rome "La Sapienza") where he obtained his degree "cum laude" in Electronical Engineering in 2005.

Then he worked on a Ph.D. in Electrical Engineering with the department of Information and Communication (INFOCOM) of the University of Rome "La Sapienza". His research interests include ICA and blind signal processing as well as adaptive filters, audio processing and neural networks for signal processing.

From 2005 to 2007 he was an associate engineer at the Central Institute for Marine Research (ICRAM), where he developed algorithms and source codes for the processing of bio-acoustic data.

From 2008 he is an assistant professor at the INFOCOM department of the University of Rome "La Sapienza", where he continues his studies on ICA, BSS and Blind Signal Processing.

He attended to several international conferences and workshops on Independent Component Analysis (ICA), Blind Signal Processing and Adaptive Signal Processing.

Index