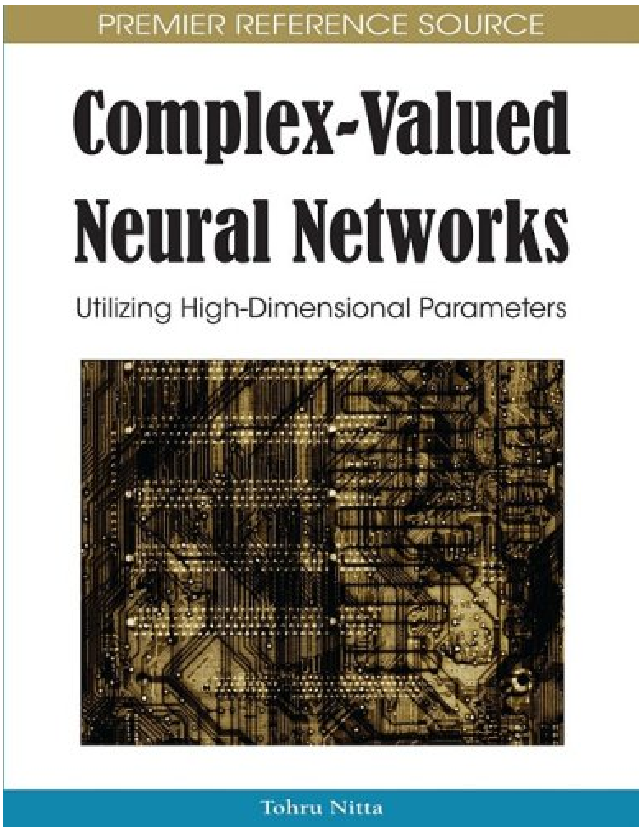
Reprinted from


PREMIER REFERENCE SOURCE
Complex-Valued Neural Networks
Utilizing High-Dimensional Parameters
Tohru Nitta

Chapter

Flexible Blind Signal Separation in the Complex Domain

Michele Scarpiniti, Daniele Vigliano, Raffaele Parisi and Aurelio Uncini,

INFO-COM Dpt.
University of Rome "La Sapienza",
Via Eudossiana, 18
00184 Roma - Italy

DOI: 10.4018/978-1-60566-214-5.ch012 ISBN13: 9781605662145
ISBN10: 1605662143
EISBN13: 9781605662152

FLEXIBLE BLIND SIGNAL SEPARATION IN THE COMPLEX DOMAIN

ABSTRACT

This chapter aims at introducing an Independent Component Analysis (ICA) approach to the separation of linear and nonlinear mixtures in complex domain. Source separation is performed by an extension of the INFOMAX approach to the complex environment. The neural network approach is based on an adaptive activation function, whose shape is properly modified during learning. Different models have been used to realize complex nonlinear functions for the linear and the nonlinear environment. In nonlinear environment the nonlinear functions involved during the learning are implemented by the so-called "*splitting functions*", working on the real and the imaginary part of the signal. In linear environment instead, the "*generalized splitting function*" which perform a more complete representation of complex function is used. Moreover a simple adaptation algorithm is derived and several experimental results are shown to demonstrate the effectiveness of the proposed method.

INTRODUCTION

In the last years *Blind Source Separation (BSS)* realized through *Independent Component Analysis (ICA)* have raised great interest in the signal processing community (Cichocki & Amari, 2002; Haykin, 2000; Roberts & Everson, 2001). In this context the neural network approach (Haykin, 1999) (usually based on a single layer perceptron (SLP) or a multilayer perceptron (MLP)) seems to be one of the preferred methodologies (Jutten & Herault, 1991; Bell & Sejnowski, 1995); 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 presented their "*neurometric architecture*" (Jutten & Herault, 1991) and Comon published his often referenced work (Comon, 1994).

Recently the problem of source separation has been extended to the complex domain (Cardoso & Laheld, 1996; Fiori et al., 1999; Bingham & Hyvärinen, 2000), due to the need of frequency domain signal processing which is quite common in telecommunication (Benvenuto et al., 1991) and biomedical applications (Calhoun et al., 2002a; Calhoun et al., 2002b). 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 (Yang & Amari, 1997). In this way one of the most important issues in designing complex neural networks consists in the definition of the complex activation function (Clarke, 1990; Benvenuto & Piazza, 1992; Kim & Adali, 2001a). In order to improve the pdf matching for the learning algorithm, the so called *Flexible ICA* was recently introduced in (Choi et al., 2000, Fiori, 2000; Solazzi et al., 2000; Vigliano & Uncini, 2003; Vigliano et al., 2005). 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 (Amari et al., 1996) and on parametric function approaches (Pham et al., 1992; Solazzi et al., 2001).

Moreover the main properties that the complex activation function should satisfy (Kim & Adali, 2002a; Vitagliano et al., 2003) 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 (Clarke, 1990; Kim & Adali, 2001a). In other words, according to this theorem, an activation function should be bounded almost everywhere in the complex domain (Clarke, 1990; Leung & Haykin, 1991; Georgiou & Koutsougeras, 1992; Kim & Adali, 2000; Kim & Adali, 2001a; Kim & Adali 2002b; Adali et al., 2004).

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, (see Jutten & Karhunen, 2003 and Vigliano et al., 2005 for an overview). In the case of the complex domain only linear instantaneous mixtures have been considered so far (Uncini et al., 1999; Bingham & Hyvärinen, 2000; Calhoun et al., 2002a; Calhoun et al., 2002b; Adali et al., 2004).

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 (Taleb & Jutten, 1999; Taleb, 2002). 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 (Uncini & Piazza, 2003). Another important issue is the definition of the theoretical conditions that grant the uniqueness of the solution.

The chapter is organized as follows: the “Background” describes the most important issues on ICA both in linear and nonlinear case. Next section (“The Complex environment”) defines the problem of separation in the complex domain. “The flexible activation function” section introduces the solution to the problem of flexibility used in this chapter, while “The de-mixing algorithm and separation architecture” defines in detail the networks used to solve the ICA problems introduced in previous sections. Finally the section “Results” describes experimental tests that demonstrate the effectiveness of the proposed approach.

BACKGROUND

Let us consider M observed signals $x_1[n],\dots,x_M[n]$ at time n, which are assumed to be the mixtures of N independent source signals $s_1[n],\dots,s_N[n]$ at time n. The vectors $\mathbf{s}[n]=\left[s_1[n],\dots,s_N[n]\right]^T$ and $\mathbf{x}[n]=\left[x_1[n],\dots,x_M[n]\right]^T$ are called the *source vector* and the *observation vector* respectively, T is 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* $\mathbf{u}[n]=\mathcal{G}\{\mathbf{x}[n]\}$ the estimate of the original source vector (see Figure 1). (Hyvärinen & Oja, 2000; Hyvärinen et al., 2001; Roberts & Everson, 2001).

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. Hence the use of the Independent Component Analysis (ICA) in BSS.

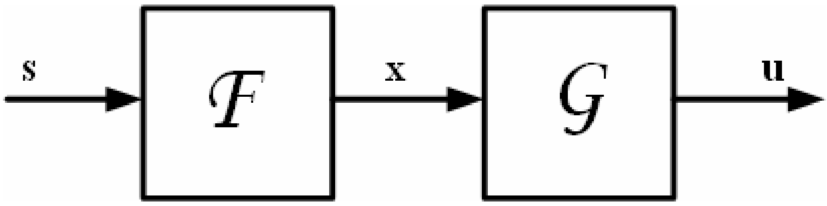


Figure 1 Model of the mixing/de-mixing system

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 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.

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 (Comon, 1994; Eriksson & Koivunen, 2004; Theis, 2004a) as an application of the Darmois-Skitovich's theorem. A consequence of this result is that the vector $\mathbf{s}$ must have at most one Gaussian component.

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

$$\mathbf{x}[n] = \mathbf{As}[n] \tag{1}$$

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

$$\mathbf{u}[n] = \mathbf{Wx}[n] \tag{2}$$

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.

Although the solution is unique, it suffers for two kinds of ambiguities (Hyvärinen & Oja, 2000; Hyvärinen et al., 2001): 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 $\mathbf{D}$ (diagonal) matrix:

$$\mathbf{WA} = \mathbf{PD} \tag{3}$$

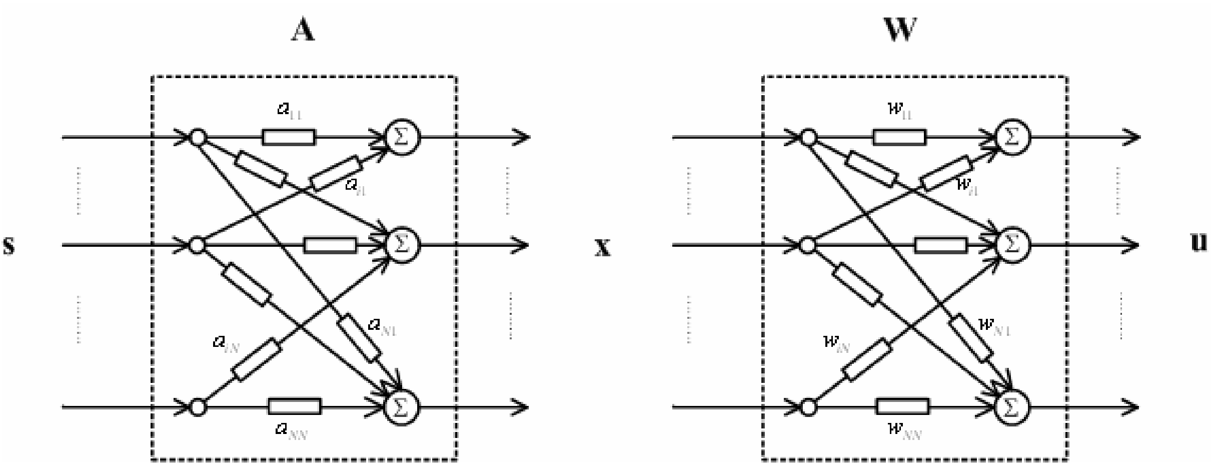


Figure 2 Mixing (left side) and de-mixing (rigth side) models

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 (Jutten & Herault, 1991; Eriksson & Koivunen, 2002; Theis & Gruber, 2005).

In order to better illustrate this aspect, consider two independent random variables s_1 with uniform distribution in $[0,\ 2\pi)$ and s_2 with Rayleigh distribution with unitary variance, so that its pdf is $p_{s_2}(s_2) = s_2 e^{-s_2^2/2}$ (Papoulis, 1991). 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 (see Jutten & Karhunen, 2003). In fact the Jacobian $\mathbf{J}$ of this transformation is:

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

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 (Theis & Gruber, 2005).

In (Taleb & Jutten, 1999) 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 3):

$$\mathbf{x}[n] = \mathcal{F}\{\mathbf{s}[n]\} = \mathbf{F}(\mathbf{A}\mathbf{s}[n]) = \mathbf{F}(\mathbf{v}[n]) \tag{4}$$

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

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 3):

$$\mathbf{u}[n] = \mathcal{G}\{\mathbf{x}[n]\} = \mathbf{W} \cdot \mathbf{G}(\mathbf{x}[n]) \tag{5}$$

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

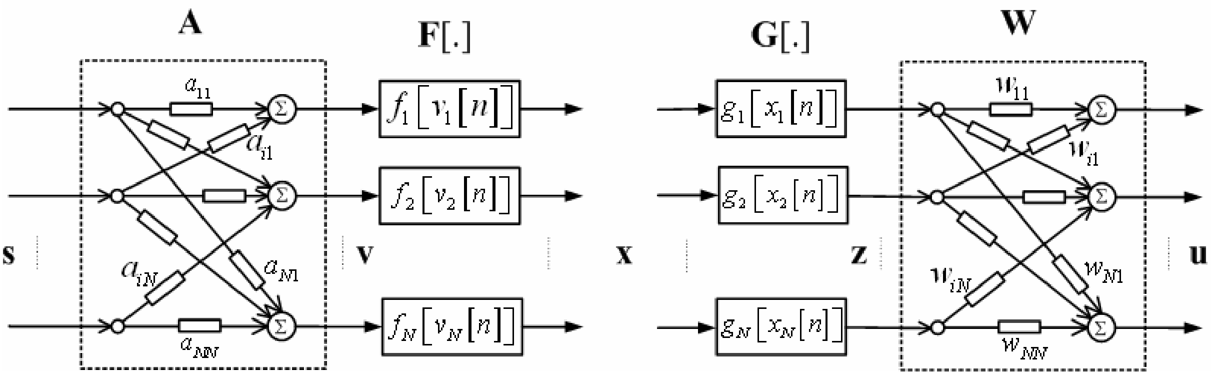


Figure 3 The PNL model: mixing model (on left side) and de-mixing mirror model (on rigth side)

Taleb & Jutten (1999) have demonstrated that if $\mathbf{A}$ and $\mathbf{W}$ are regular matrices, $f_i(v_i)$ ($\forall i = 1,2,\ldots,N$) 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 Taleb & Jutten (1999) contains an inaccuracy which was solved by Theis & Gruber (2005).

THE COMPLEX ENVIRONMENT

The aim of this section is to extend the BSS problem introduced above to the complex domain. Let us consider a vector $\mathbf{s}[n] = \left[s_1[n],\ldots,s_N[n]\right]^T$ 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. The goal of BSS is to recover the complex signal $\mathbf{s}[n]$ from observations of the complex mixture $\mathbf{x}[n] = \left[x_1[n],\ldots,x_N[n]\right]^T$, 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 (1) and (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 $w_{ij} \in \mathbb{C}$):

$$\mathbf{x}[n] = \mathbf{A}\mathbf{s}[n] \tag{6}$$

for the mixing model and

$$\mathbf{u}[n] = \mathbf{W}\mathbf{x}[n] \tag{7}$$

for the de-mixing model (see Figure 2).

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

$$\mathbf{x}[n] = \mathbf{x}_R[n] + j\mathbf{x}_I[n] = \mathbf{F}(\mathbf{v}[n]) \tag{8}$$

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 $\mathbf{F}(\mathbf{v}[n]) = \left[f_1(v_1[n]), \cdots, f_N(v_N[n]) \right]^T$, where $f_k(v_k[n])$ is the *k*-th complex nonlinear distorting function. For the mirror de-mixing model:

$$\mathbf{u}[n] = \mathbf{u}_R[n] + j\mathbf{u}_I[n] = \mathbf{W}\mathbf{r}[n] = \mathbf{W} \cdot \mathbf{G}(\mathbf{x}[n]) \tag{9}$$

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 $\mathbf{G}\{\mathbf{x}[n]\} = \left[g_1(x_1[n]), \cdots, g_N(x_N[n]) \right]^T$, where $g_k(x_k[n])$ is the *k*-th complex nonlinear compensating function.

The sparability, identifiability and uniqueness of the solution is demonstrated in Theis (2004b) and Eriksson & Koivunen (2006) in the linear and instantaneous case, while the PNL mixtures is not treated yet in literature. It is intention of this chapter to extend these results in the nonlinear environment.

The Complex Nonlinear Activation Function

One of the main issues in designing complex neural networks is the presence of complex nonlinear functions involved in the learning processing (Pham et al., 1992; Calhoun et al., 2002b), i.e. complex activation functions or distorting functions (for the nonlinear mixing environment).

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 part of the complex variable z and $j = \sqrt{-1}$ is the imaginary unit.

The main challenge is the dichotomy between boundedness and analyticity in the complex domain (Kim & Adali, 2002a), as stated by the *Liouville's theorem: 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 (Georgiou & Koutsougeras, 1992) 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: $|h(z)| \le c < \infty$
3. $h^R_{z_R}, h^R_{z_I}, h^I_{z_R}, h^I_{z_I}$ exist and are bounded (10)
4. $h(z)$ is not entire[1]
5. $h^R_{z_R} h^I_{z_I} \neq h^R_{z_I} h^I_{z_R}$

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

where $h^R\left(z_R,z_I\right)$ and $h^I\left(z_R,z_I\right)$ are known as the real part function and imaginary part function of the complex function $h\left(z\right)$ 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 (10) require the boundedness of the nonlinear function and its derivatives even when the function is defined in a local domain (Calhoun & Adali, 2002). 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 (Kim & Adali, 2001b; 2002b). 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 (1989) to the complex case.

In this context Kim & Adali (Kim & Adali, 2001a; 2002a) proposed 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 $h\left(z\right)$:

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

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

Recently Adali et al. (2004) have used these above ETFs as complex activation functions in BSS problems. In particular, the functions: $tan^{-1}\left(z\right)$, $sin^{-1}\left(z\right)$, $cos^{-1}\left(z\right)$ and $tan\left(z\right)$, and their hyperbolic counterparts performed consistently well over a wide range of input and mixtures, while the functions: $sin\left(z\right)$, $cos\left(z\right)$, $cot^{-1}\left(z\right)$, $sinh\left(z\right)$, $cosh\left(z\right)$ and $coth^{-1}\left(z\right)$ exhibited unstable behaviour when used for ICA with the algorithm proposed by Bell & Sejnowski (1995) and described later.

According to the properties (10) 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 (see Benvenuto et al., 1991 and further Smaragdis, 1998 and Uncini & Piazza, 2003). The splitting function

$$h\left(z\right)=h\left(z_R,z_I\right)=h^R\left(z_R\right)+jh^I\left(z_I\right) \tag{11}$$

[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.

avoids the problem of unboundedness of complex nonlinearities, as stated above, but it cannot be analytic (see Figure 4).

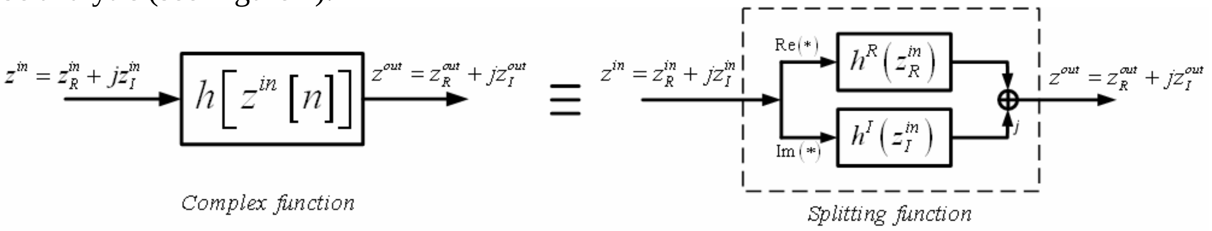

$z^{in} = z_R^{in} + jz_I^{in}$
$h\left[z^{in}[n]\right]$
$z^{out} = z_R^{out} + jz_I^{out}$
≡
$z^{in} = z_R^{in} + jz_I^{in}$
Re(*)
Im(*)
$h^R\left(z_R^{in}\right)$
$h^I\left(z_I^{in}\right)$
j
$z^{out} = z_R^{out} + jz_I^{out}$
Complex function
Splitting function

Figure 4 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 & al., 2003 proposed a complex neural network based on a couple of bi-dimensional functions (Figure 5) called *generalized splitting function*:

$$h(z) = h(z_R, z_I) = h^R(z_R, z_I) + jh^I(z_R, z_I) \tag{12}$$

in this way $h(z)$ is bounded and analytic if one imposes the *Cauchy-Riemann condition*s ($h_{z_R}^R = h_{Z_I}^I$, $h_{z_R}^I = -h_{Z_I}^R$). Note that the Cauchy-Riemann conditions are equivalent to the fifth property in (10).

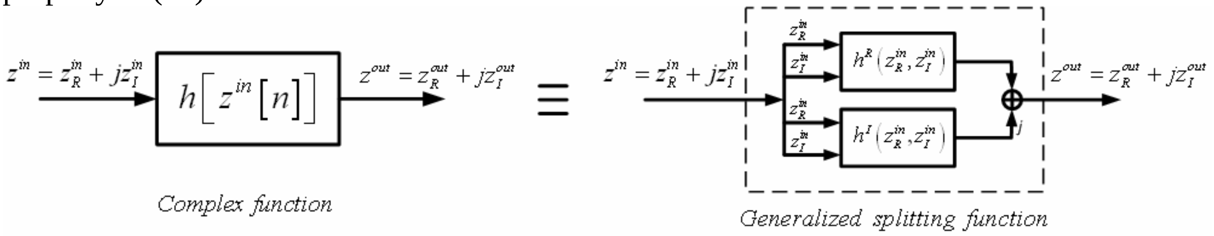

$z^{in} = z_R^{in} + jz_I^{in}$
$h\left[z^{in}[n]\right]$
$z^{out} = z_R^{out} + jz_I^{out}$
≡
$z^{in} = z_R^{in} + jz_I^{in}$
z_R^{in}
z_I^{in}
z_R^{in}
z_I^{in}
$h^R\left(z_R^{in}, z_I^{in}\right)$
$h^I\left(z_R^{in}, z_I^{in}\right)$
j
$z^{out} = z_R^{out} + jz_I^{out}$
Complex function
Generalized splitting function

Figure 5 Generalized splitting function realization of a complex activation function (AF)

Another approach for adapting the nonlinearity to the source distribution is introduced in Adali & Li, (2007). It is based on the idea that the simple substitutions $z_R = \left(z + z^*\right)/2$ and $z_I = \left(z - z^*\right)/2j$ (where $(\bullet)^*$ denotes the complex conjugate) allow us to write a given pdf that is $\mathbb{R}\times\mathbb{R}\rightarrow\mathbb{R}$ in terms of a function $\mathbb{C}\times\mathbb{C}\rightarrow\mathbb{R}$.

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 (Taleb, 2002).

The model of nonlinear complex compensating functions $\mathbf{G}\left(\mathbf{x}[n]\right)$ and the activation functions (AF) considered in this chapter are realized by *splitting function* according to eq. (11) (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 3) having independent components u_i with marginal pdf $p_{u_i}(u_i)$

$$\mathcal{U} = \left\{\mathbf{u}\,\middle|\, p_{\mathbf{u}}(\mathbf{u}) = \prod_i p_{u_i}(u_i);\;\; \mathbf{u} = \mathcal{G}\{\mathbf{x}\} = \mathcal{G}\circ\mathcal{F}\{\mathbf{s}\} = \mathcal{H}\{\mathbf{s}\}\right\} \tag{13}$$

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 $\mathbf{u} = \mathcal{G}\{\mathbf{x}\} \in \mathcal{V}$, but not all of them have a diagonal Jacobian matrix. So most of the solutions in $\mathcal{V}$ 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 $\mathbf{F}(\mathbf{v}[n]) = \mathbf{F}_R(\mathbf{v}_R[n]) + j\mathbf{F}_I(\mathbf{v}_I[n])$, $\mathbf{F}_R(\mathbf{v}_R[n]) = \left[f_{R1}(v_{R1}[n]),\ldots, f_{RN}(v_{RN}[n])\right]^T$ and $\mathbf{F}_I(\mathbf{v}_I[n]) = \left[f_{I1}(v_{I1}[n]),\ldots, f_{IN}(v_{IN}[n])\right]^T$ the complex domain mixing environment (8), represented in Vigliano et al. (2006a) and Vigliano et al.(2008), can be rewritten in the following way:

$$\begin{aligned}\tilde{\mathbf{x}}[n] &= \begin{bmatrix} \mathbf{x}_R[n] \\ \mathbf{x}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{F}_R\{\mathbf{v}_R[n]\} \\ \mathbf{F}_I\{\mathbf{v}_I[n]\} \end{bmatrix} \\ \tilde{\mathbf{v}}[n] &= \begin{bmatrix} \mathbf{v}_R[n] \\ \mathbf{v}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R \end{bmatrix} \begin{bmatrix} \mathbf{s}_R[n] \\ \mathbf{s}_I[n] \end{bmatrix} = \tilde{\mathbf{A}}\tilde{\mathbf{s}}\end{aligned} \tag{14}$$

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 (14) 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. (9) 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 (Vigliano et al., 2005; 2008).

The nonlinear complex compensating functions $\mathbf{G}(\bullet)$ have been realized as splitting functions according to eq. (11): $\mathbf{G}(\mathbf{x}_R[n] + j\mathbf{x}_I[n]) = \mathbf{G}_R(\mathbf{x}_R[n]) + j\mathbf{G}_I(\mathbf{x}_I[n])$.

Similarly to eq. (14) it is possible to express the complex de-mixing model (9) by using real expressions only:

$$\tilde{\mathbf{u}} = \begin{bmatrix} \mathbf{u}_R[n] \\ \mathbf{u}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix} \begin{bmatrix} \mathbf{G}_R\{\mathbf{x}_R[n]\} \\ \mathbf{G}_I\{\mathbf{x}_I[n]\} \end{bmatrix} = \tilde{\mathbf{W}} \cdot \tilde{\mathbf{G}}[\tilde{\mathbf{x}}] \tag{15}$$

in which $\mathbf{G}_R(\mathbf{x}_R[n]) = \left[g_{R1}(x_{R1}[n]),\cdots, g_{RN}(x_{RN}[n])\right]^T$ and $\mathbf{G}_I(\mathbf{x}_I[n]) = \left[g_{I1}(x_{I1}[n]),\cdots, g_{IN}(x_{IN}[n])\right]^T$ are the real and imaginary part 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 (15) 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 (Taleb & Jutten, 1999) for the real PNL case (already applied in Vigliano et al., 2005 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 $\mathcal{V}$, under proper constraints, differ only for a trivial ambiguity if the mixing model is eq. (8) and the de-mixing model is eq. (9). This is shown in the following proposition.

Proposition 1 *Given the nonlinear complex mixing model* $\mathcal{F}\{\mathbf{A},\mathbf{F}\}$ *in eq. (8) and the recovery model* $\mathcal{G}\{\mathbf{G},\mathbf{W}\}$ *in eq. (9), let us assume that:*

(a) $\mathbf{A}$ *is a non-singular matrix of non zero entries (both for real and imaginary part);*

(b) $f_{Ri}(\bullet)$, $f_{Ii}(\bullet)$, $g_{Ri}(\bullet)$, $g_{Ii}(\bullet)$ *(*$i=1\ldots N$*) are 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:*

$$\mathbf{u}[n]=\mathbf{P\Lambda s}[n]=\mathbf{P}\begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_N\end{bmatrix}\mathbf{s}[n] \qquad (16)$$

In eq. (16) $\mathbf{P}$ *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.

Proposition 1 ensures the existence and uniqueness of the solution at the expense of strong constraints on the real and the imaginary parts of the signals.

THE FLEXIBLE ACTIVATION FUNCTION

A second problem in choosing complex AF is the matching between the shape of the AF and the cumulative density function (cdf) of the unknown sources (Yang & Amari, 1997). The idea is to adopt a flexible solution (Choi et al., 2000): the shape of the AF is adaptively changed from data by the use of flexible functions, performing the so-called *Flexible ICA*[3]. This solution allows the separation of signal with a no pre-defined cumulative density function (cdf) and moreover increases the quality of the separation.

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 (Uncini et al., 1999). The digital implementation of this kind of activation function through a look-up-table (LUT) keeps the complexity under control (Piazza et al., 1993) 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 (Benvenuto et al., 1991; Guarnieri et al., 1999; Uncini et al., 1999; Solazzi et al., 2000; Solazzi & Uncini, 2000).

[3] *Flexible ICA* defines an ICA algorithm having flexible activation functions which are able to match adaptively the cumulative density function (cdf) of the original signals.

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) (Catmull & Rom, 1974; Vecci et al., 1998; Uncini et al., 1999; Uncini & Piazza, 2003). There is a regularization property common to most of the polynomial spline basis sets called variation diminishing property, 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. If we uniformly sample the abscissas along the *u*-axis, $\Delta u = q_{u,i+1} - q_{u,i} = \text{const}$ for every i, then the cubic polynomial becomes a first degree polynomial.

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*. A comparison between these two schemes is shown in left side of Figure 6 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. (18).

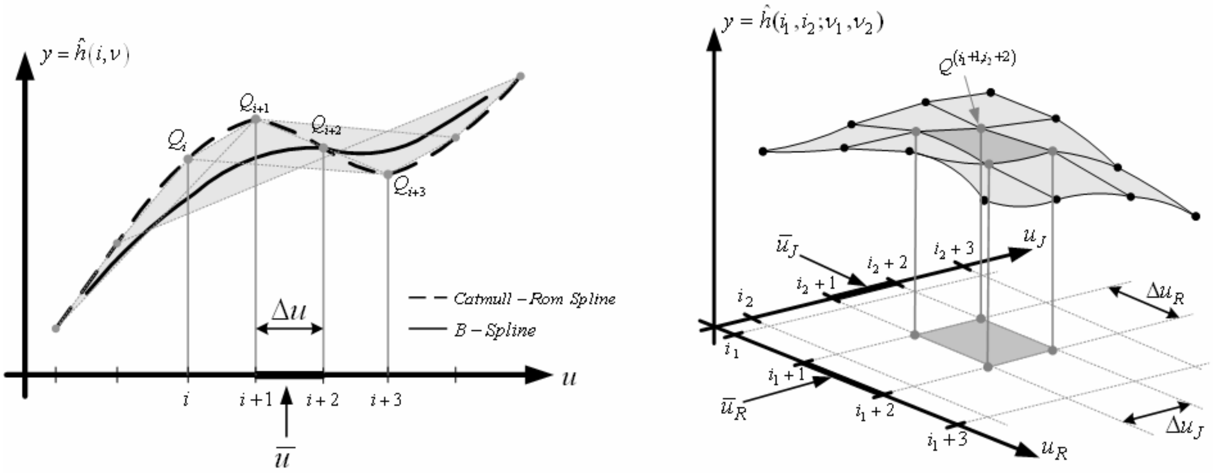


Figure 6 Mono and Bi-dimensional cubic spline functions

A common expression of these spline activation functions, is the following matrix notation:

$$y = \hat{h}(i,\nu) = \mathbf{T}_\nu \cdot \mathbf{M} \cdot \mathbf{Q}_i \tag{17}$$

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

$$\mathbf{T}_\nu = \begin{bmatrix} \nu^3 & \nu^2 & \nu & 1 \end{bmatrix},\ \mathbf{M} = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{Q}_i = \begin{bmatrix} Q_i \\ Q_{i+1} \\ Q_{i+2} \\ Q_{i+3} \end{bmatrix} \tag{18}$$

and where $Q_i = \begin{bmatrix} q_{u,i} & q_{y,i} \end{bmatrix}^T$, and $0 \le \nu \le 1$ is the local abscissa.

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:

$$\begin{aligned}
&\bar{\zeta} = \frac{\bar{u}}{\Delta u} + \frac{N-2}{2} \\
&\zeta = \begin{cases} 1 & \bar{\zeta} < 1 \\ \bar{\zeta} & 1 \le \bar{\zeta} \le N-3 \\ N-3 & \bar{\zeta} > N-3 \end{cases} \\
&i = \lfloor \zeta \rfloor \\
&\nu = \zeta - i
\end{aligned} \tag{19}$$

where $\lfloor \bullet \rfloor$ is the floor operator and N is the number of spline control points.

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. 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 (Solazzi et al., 2000; Solazzi & Uncini, 2000; 2004). 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 $0 \le \nu_R, \nu_I \le 1$, and in matrix formulation it is expressed as follows:

$$y = \hat{h}(i_R, i_I; \nu_R, \nu_I) = \mathbf{T}_{\nu I} \cdot \mathbf{M} \cdot (\mathbf{T}_{\nu R} \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]}^{(i_R,i_I)})^T \tag{20}$$

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 (18) and $\mathbf{Q}_{[2]}^{(i_R,i_I)}$ is a bi-dimensional structure collected the local control points:

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

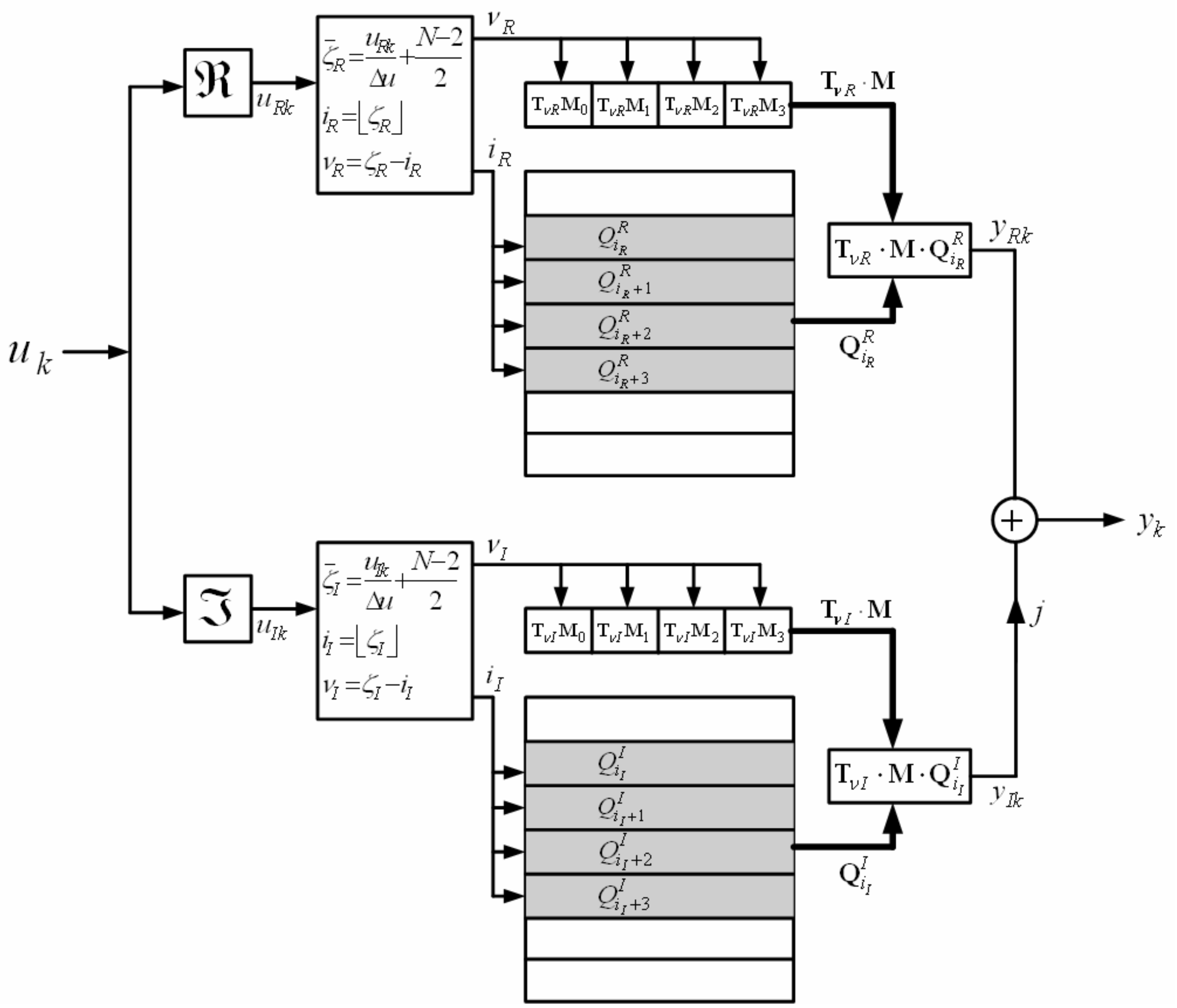
An example of cubic 2D spline is showed in right side of Figure 6.

The new idea is to use spline-based functions for $h^R(\bullet)$ and $h^I(\bullet)$ in eq. (11). Thus, remembering the spline matrix formulation in eq. (17) 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_J, as stated by equations (19), then the AF in eq. (11) can be written as follows using eq. (17):

$$y_k = h_k^R(u_{Rk}) + jh_k^I(u_{Ik}) = \mathbf{T}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}_{i_R}^R + j\mathbf{T}(u_{Ik}) \cdot \mathbf{M} \cdot \mathbf{Q}_{i_I}^I \tag{21}$$

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. The data path is reported in Figure 7.

The corresponding neural network is called *Complex valued Adaptive Spline Neural Network (CASNN).*



u_k
$\Re$
u_{Rk}
$\bar{\zeta}_R = \frac{u_{Rk}}{\Delta u} + \frac{N-2}{2}$
$i_R = \lfloor \zeta_R \rfloor$
$v_R = \zeta_R - i_R$
v_R
i_R
$T_{vR}M_0$ $T_{vR}M_1$ $T_{vR}M_2$ $T_{vR}M_3$
$\mathbf{T}_{vR} \cdot \mathbf{M}$
$Q_{i_R}^R$
$Q_{i_R+1}^R$
$Q_{i_R+2}^R$
$Q_{i_R+3}^R$
$\mathbf{Q}_{i_R}^R$
$\mathbf{T}_{vR} \cdot \mathbf{M} \cdot \mathbf{Q}_{i_R}^R$
y_{Rk}
y_k
j
$\Im$
u_{Ik}
$\bar{\zeta}_I = \frac{u_{Ik}}{\Delta u} + \frac{N-2}{2}$
$i_I = \lfloor \zeta_I \rfloor$
$v_I = \zeta_I - i_I$
v_I
i_I
$T_{vI}M_0$ $T_{vI}M_1$ $T_{vI}M_2$ $T_{vI}M_3$
$\mathbf{T}_{vI} \cdot \mathbf{M}$
$Q_{i_I}^I$
$Q_{i_I+1}^I$
$Q_{i_I+2}^I$
$Q_{i_I+3}^I$
$\mathbf{Q}_{i_I}^I$
$\mathbf{T}_{vI} \cdot \mathbf{M} \cdot \mathbf{Q}_{i_I}^I$
y_{Ik}

Figure 7 Mono-dimensional spline data path of a complex activation function (AF)

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 & al. (2003) proposed a complex neural network based on bi-dimensional spline AF.

If we consider the expression (12) 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)* (Vitagliano et al, 2003; Scarpiniti et al., 2007).

Using the compact matrix formulation in eq. (20) we have for the *k*-th AF:

$$\begin{aligned} y_{Rk} &= h_k^R(u_{Rk}, u_{Ik}) = \mathbf{T}_{vI}(u_{Ik}) \cdot \mathbf{M} \cdot (\mathbf{T}_{vR}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]R}^{(i_R, i_R+3)})^T \\ y_{Ik} &= h_k^I(u_{Rk}, u_{Ik}) = \mathbf{T}_{vI}(u_{Ik}) \cdot \mathbf{M} \cdot (\mathbf{T}_{vR}(u_{Rk}) \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]I}^{(i_I, i_I+3)})^T \\ y_k &= y_{Rk} + jy_{Ik} \end{aligned} \tag{22}$$

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 8.

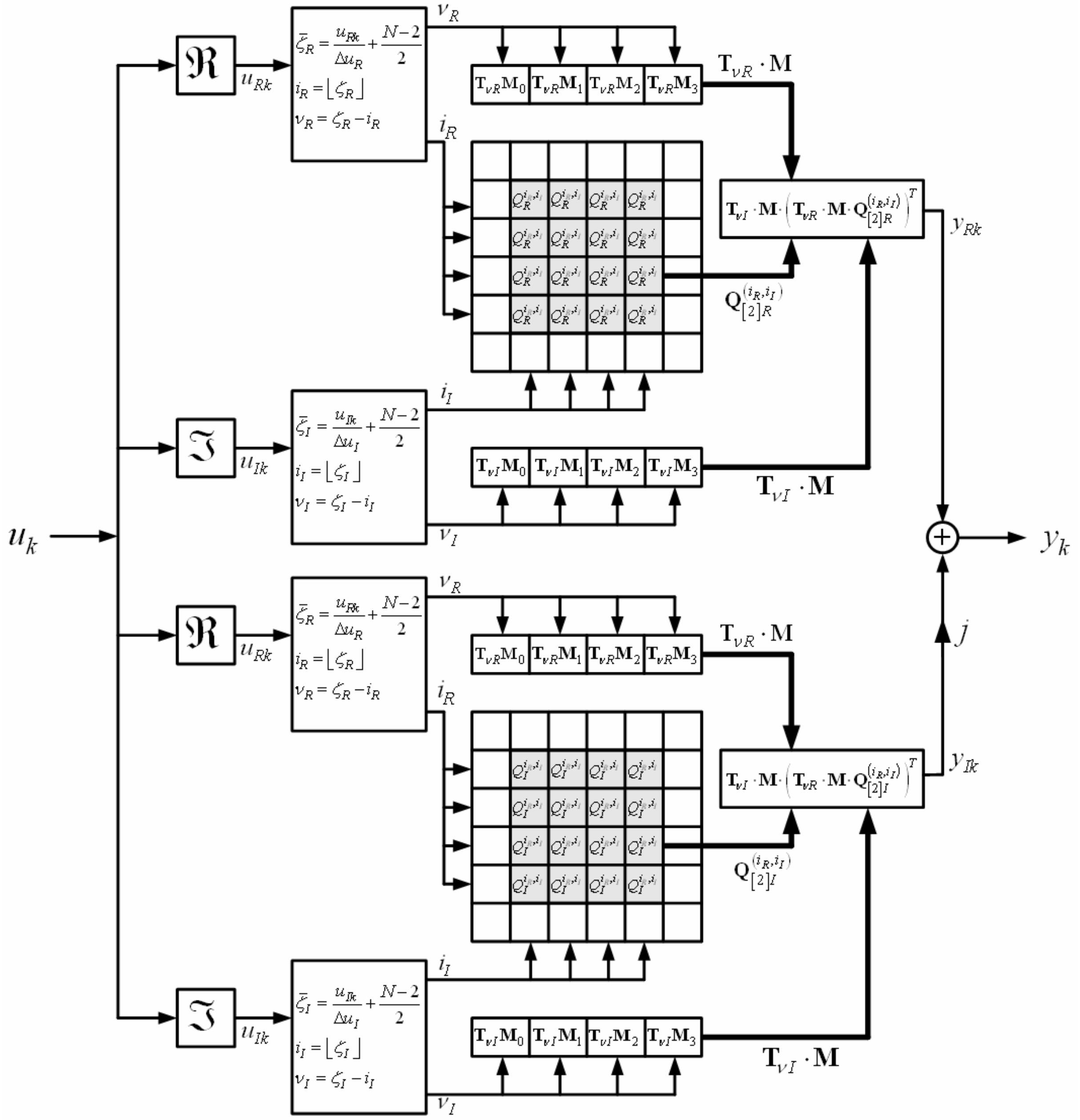

u_k
$\Re$
$\Im$
u_{Rk}
u_{Ik}
$\zeta_R = \frac{u_{Rk}}{\Delta u_R} + \frac{N-2}{2}$
$i_R = \lfloor \zeta_R \rfloor$
$v_R = \zeta_R - i_R$
$\zeta_I = \frac{u_{Ik}}{\Delta u_I} + \frac{N-2}{2}$
$i_I = \lfloor \zeta_I \rfloor$
$v_I = \zeta_I - i_I$
v_R
i_R
i_I
v_I
$\mathbf{T}_{vR}\mathbf{M}_0$ $\mathbf{T}_{vR}\mathbf{M}_1$ $\mathbf{T}_{vR}\mathbf{M}_2$ $\mathbf{T}_{vR}\mathbf{M}_3$
$\mathbf{T}_{vI}\mathbf{M}_0$ $\mathbf{T}_{vI}\mathbf{M}_1$ $\mathbf{T}_{vI}\mathbf{M}_2$ $\mathbf{T}_{vI}\mathbf{M}_3$
$\mathbf{T}_{vR} \cdot \mathbf{M}$
$\mathbf{T}_{vI} \cdot \mathbf{M}$
$Q_R^{i_R,i_I}$
$Q_I^{i_R,i_I}$
$\mathbf{T}_{vI} \cdot \mathbf{M} \cdot \left(\mathbf{T}_{vR} \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]R}^{(i_R,i_I)}\right)^T$
$\mathbf{T}_{vI} \cdot \mathbf{M} \cdot \left(\mathbf{T}_{vR} \cdot \mathbf{M} \cdot \mathbf{Q}_{[2]I}^{(i_R,i_I)}\right)^T$
$\mathbf{Q}_{[2]R}^{(i_R,i_I)}$
$\mathbf{Q}_{[2]I}^{(i_R,i_I)}$
y_{Rk}
y_{Ik}
j
y_k

Figure 8 Bi-dimensional spline data path of a complex activation function (AF)

THE DE-MIXING ALGORITHM AND SEPARATION ARCHITECTURE

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 the Introduction section 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 & Sejnowski (1995). This learning algorithm maximizes information transferred by the nonlinear network shown in Figure 9, assuming no knowledge on input vector distribution.

INFOMAX addresses the problem of maximizing the mutual information $I(\mathbf{y},\mathbf{x})$ (Cover & Tomas, 2006), between the input vector $\mathbf{x}$ and an invertible nonlinear transform of it, $\mathbf{y}$ obtained as

$$\mathbf{y}=\mathbf{h}(\mathbf{u})=\mathbf{h}(\mathbf{W}\mathbf{x}) \tag{23}$$

where $\mathbf{W}$ is an $N\times N$ matrix and $\mathbf{h}(\mathbf{u})=\left[h_1(u_1),\dots,h_N(u_N)\right]^T$ is the nonlinear function vector (see the sixth chapter in Haykin, 2000). Because the mapping in eq. (23) 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:

$$I(\mathbf{y},\mathbf{x})=H(\mathbf{y})-H(\mathbf{y}|\mathbf{x}) \tag{24}$$

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 eq. (24). In order to derive the learning algorithm let we pose $p_{\mathbf{x}}(\mathbf{x})$ and $p_{\mathrm{y}}(\mathbf{y})$ the probability density functions (pdf) of the network input and output respectively which have to satisfy the relation (Papoulis, 1991):

$$p_{\mathrm{y}}(\mathbf{y})=\frac{p_{\mathrm{x}}(\mathbf{x})}{|\det\mathbf{J}|} \tag{25}$$

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

Since the joint entropy of network output is defined as $H(\mathbf{y})=-E\left\{\ln p_{\mathrm{y}}(\mathbf{y})\right\}$ (Cover & Tomas, 2006), where $E\{\bullet\}$ is the expected value operator, substituting into it the eq. (25) we obtain:

$$H(\mathbf{y})=E\left\{\ln|\det\mathbf{J}|\right\}+H(\mathbf{x}) \tag{26}$$

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

$$\ln|\det\mathbf{J}|=\ln\det\mathbf{W}+\sum_{i=1}^{N}\ln|h_i'| \tag{27}$$

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

$$H(\mathbf{y})=H(\mathbf{x})+\ln\det\mathbf{W}+\sum_{i=1}^{N}\ln|h_i'| \tag{28}$$

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

$$\mathbf{\Phi}(l+1)=\mathbf{\Phi}(l)+\eta_{\mathbf{\Phi}}\frac{\partial\mathcal{L}\{\mathbf{\Phi}(l)\}}{\partial\mathbf{\Phi}}=\mathbf{\Phi}(l)+\eta_{\mathbf{\Phi}}\Delta\mathbf{\Phi}(l) \tag{29}$$

where η_{Φ} 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 $\mathbf{W}$ using the stochastic gradient method in eq. (29) as follows:

$$\Delta\mathbf{W} = \frac{\partial H(\mathbf{y})}{\partial \mathbf{W}} = \mathbf{W}^{-T} + \mathbf{\Psi}\mathbf{x}^{T} \tag{30}$$

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

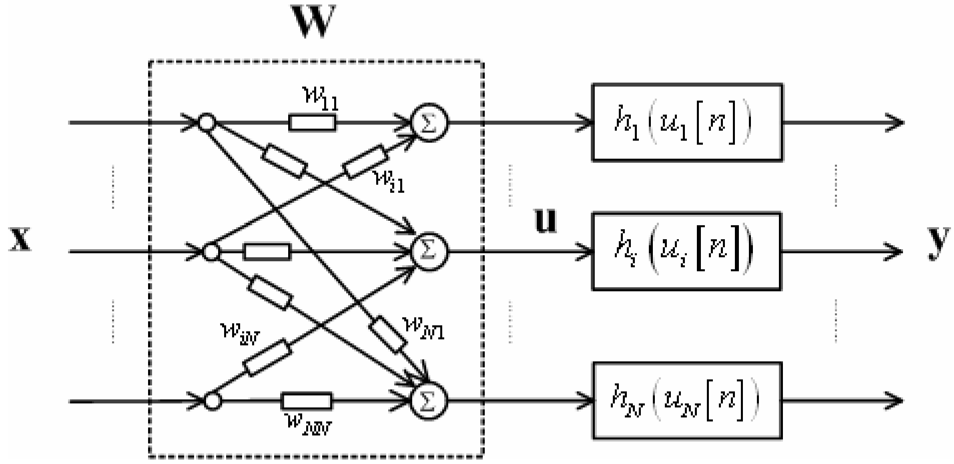


Figure 9 The nonlinear network used to introduce the INFOMAX principle

Moreover we can introduce the mutual information of the linear outputs $\mathbf{u}$ as the Kullback-Leibler distance (Cover & Tomas, 2006) 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 eq. (27) in the definition of entropy, after some easy passes, we obtain:

$$H(\mathbf{y}) = -I(\mathbf{u}) + E\left\{\sum_{i=1}^{N} \ln \frac{|h_i'|}{p_{u_i}(u_i)}\right\} \tag{31}$$

Thus if $|h_i'| = p_{u_i}(u_i)$ ($\forall i$) then maximizing the joint entropy $H(\mathbf{y})$ can solve the ICA problem. In this way $h_i(u_i)$ should be the cumulative density function (cdf) of the *i*-th estimated source. Hence the importance to use an adaptive AF.

From eq. (31) 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*) (Yang & Amari, 1997; Theis et al., 2002).

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 (Calhoun & Adali, 2002). 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 (Lee et al., 2000).

The Case of Linear Environment

The architecture used to realize the model $\mathbf{W}$ in eq. (7) is represented in Figure 10a.

Let we consider first the case of using the splitting activation function in eq. (11) realized through the mono-dimensional spline activation function in eq. (21), assuming

$$\begin{aligned} \mathbf{y} &= \mathbf{h}(\mathbf{u}) = \mathbf{h}_R(\mathbf{u}_R) + j\mathbf{h}_I(\mathbf{u}_I) \\ y_k &= y_{Rk} + jy_{Ik} = h_{Rk}(u_{Rk}) + jh_{Ik}(u_{Ik}) \end{aligned} \tag{32}$$

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 (see eqs. (14) and (15)) by using only real terms:

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_R[n] \\ \mathbf{y}_I[n] \end{bmatrix} = \begin{bmatrix} \mathbf{h}_R\left\{\mathbf{u}_R[n]\right\} \\ \mathbf{h}_I\left\{\mathbf{u}_I[n]\right\} \end{bmatrix} \tag{33}$$

In this way $\tilde{\mathbf{y}}$ is a real vector of $2N$ elements. Considering a de-mixing model with parameters $\mathbf{\Phi} = \left\{w_{ij}, \mathbf{\Phi}_h \;\middle|\forall i, j\;\right\}$, 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 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. (26):

$$\mathcal{L}\left\{\mathbf{y}[n], \mathbf{\Phi}\right\} = H\left(\tilde{\mathbf{y}}\right) = -E\left\{\ln\left(p_{\tilde{\mathbf{y}}}\left(\tilde{\mathbf{y}}\right)\right)\right\} = H\left(\tilde{\mathbf{x}}\right) + E\left\{\ln\left(\tilde{\mathbf{J}}\right)\right\} \tag{34}$$

In eq. (34) the output pdf $p_{\tilde{\mathbf{y}}}\left(\tilde{\mathbf{y}}\right)$ can be expressed using eq. (25) as a function of the model's parameters and of $H\left(\tilde{\mathbf{x}}\right)$ 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:

$$\det\left(\tilde{\mathbf{J}}\right) = \prod_{k=1}^{N} \dot{\tilde{y}}_k \det\left(\tilde{\mathbf{W}}\right) \tag{35}$$

In eq. (35) $\dot{\tilde{y}}_k$ 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. (35) in eq. (34) we obtain the cost function whose derivation with respect the elements of $\mathbf{\Phi}$ leads to

$$\frac{\partial}{\partial \mathbf{\Phi}} \mathcal{L}\left\{\mathbf{y}, \mathbf{\Phi}\right\} = \frac{\partial}{\partial \mathbf{\Phi}} \left[\ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) + \sum_{k=1}^{N} \ln \dot{y}_{Rk} + \sum_{k=1}^{N} \ln \dot{y}_{Ik} \right] \tag{36}$$

where $\dot{y}_{Rk}$ and $\dot{y}_{Ik}$ denote the derivative of the *k*-th real and imaginary part of the network output $\mathbf{y}$. In eq. (36) expected values have been replaced by instantaneous values.

Maximization of eq. (36) by the stochastic gradient method in eq. (29) yields three learning rules (see Uncini & Piazza (2003) for major details). The learning rule for the network's weights is:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \mathbf{\Psi}\mathbf{x}^H \tag{37}$$

where H is the Hermitian operator, $\mathbf{W}^{-H} = \left(\mathbf{W}^{-1}\right)^H$, $\mathbf{\Psi} = \mathbf{\Psi}_R + j\mathbf{\Psi}_I$, $\boldsymbol{\psi}_R = \left[\Psi_{R1}, \ldots, \Psi_{RN}\right]^T$, $\boldsymbol{\psi}_{kI} = \left[\Psi_{I1}, \ldots, \Psi_{IN}\right]^T$, $\Psi_{Rk} = \ddot{y}_{Rk} / \dot{y}_{Rk}$, $\Psi_{Ik} = \ddot{y}_{Ik} / \dot{y}_{Ik}$, $\dot{y}_{Rk}$ 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. (17) the terms Ψ_{Rk} and Ψ_{Ik} can be expressed as follows:

$$\begin{aligned} \Psi_{Rk} &= \frac{1}{\Delta u_R} \frac{\ddot{\mathbf{T}}_{\nu R}\mathbf{M}\mathbf{Q}_R^h}{\dot{\mathbf{T}}_{\nu R}\mathbf{M}\mathbf{Q}_R^h} \\ \Psi_{Ik} &= \frac{1}{\Delta u_I} \frac{\ddot{\mathbf{T}}_{\nu I}\mathbf{M}\mathbf{Q}_I^h}{\dot{\mathbf{T}}_{\nu I}\mathbf{M}\mathbf{Q}_I^h} \end{aligned} \tag{38}$$

where $\dot{\mathbf{T}}_{\nu R} = \begin{bmatrix} 3\nu_R^2 & 2\nu_R & 1 & 0 \end{bmatrix}$, $\ddot{\mathbf{T}}_{\nu R} = \begin{bmatrix} 6\nu_R & 2 & 0 & 0 \end{bmatrix}$ 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:

$$\begin{aligned} \Delta\mathbf{Q}^{h}_{\mathrm{R},k,i+m} &= \frac{\dot{\mathbf{T}}_{\nu R}\left(u_{\mathrm{R}k}\right)\left(\mathbf{M}\right)_{m}}{\dot{\mathbf{T}}_{\nu R}\left(u_{\mathrm{R}k}\right)\mathbf{M}\mathbf{Q}^{h}_{\mathrm{R},k,i+m}} \\ \Delta\mathbf{Q}^{h}_{\mathrm{I},k,i+m} &= \frac{\dot{\mathbf{T}}_{\nu I}\left(u_{\mathrm{I}k}\right)\left(\mathbf{M}\right)_{m}}{\dot{\mathbf{T}}_{\nu I}\left(u_{\mathrm{I}k}\right)\mathbf{M}\mathbf{Q}^{h}_{\mathrm{I},k,i+m}} \end{aligned} \tag{39}$$

where $\left(\mathbf{M}\right)_{m}$ is the *m*-th column of the $\mathbf{M}$ matrix.

We can generalize this algorithm using the generalized splitting function in eq. (12) realizing the complex AFs with the *bi-dimensional spline function* in eq. (22) (Scarpiniti et al., 2007; 2008) shown in Figure 10b. 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. (37), but the generic *k*-th terms Ψ_{Rk} and Ψ_{Ik} of the vector $\boldsymbol{\Psi}$ 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}.

The learning rules for the real activation function become

$$\begin{aligned} \Delta\mathbf{Q}^{h}_{\mathrm{R},k,i_R+m_R,i_I+m_I} = 2\Bigg(& \frac{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}_{m_J}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}_{m_R}\right)^{T}\right)}{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)^{2}+\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)^{2}}+ \\ & +\frac{\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}_{m_J}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}_{m_R}\right)^{T}\right)}{\left(\mathbf{T}_{\nu I}\cdot\mathbf{M}\cdot\left(\dot{\mathbf{T}}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)^{2}+\left(\dot{\mathbf{T}}_{\nu I}\cdot\mathbf{M}\cdot\left(\mathbf{T}_{\nu R}\cdot\mathbf{M}\cdot\mathbf{Q}^{(i_R,i_I)}_{[2]R}\right)^{T}\right)^{2}}\Bigg) \end{aligned} \tag{40}$$

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 $\mathbf{M}$. A similar equation can be result for the imaginary surface $\Delta\mathbf{Q}^{h}_{\mathrm{I},k,i_R+m_R,i_J+m_J}$. For a complete derivation of the learning rules see (Scarpiniti et al., 2008).

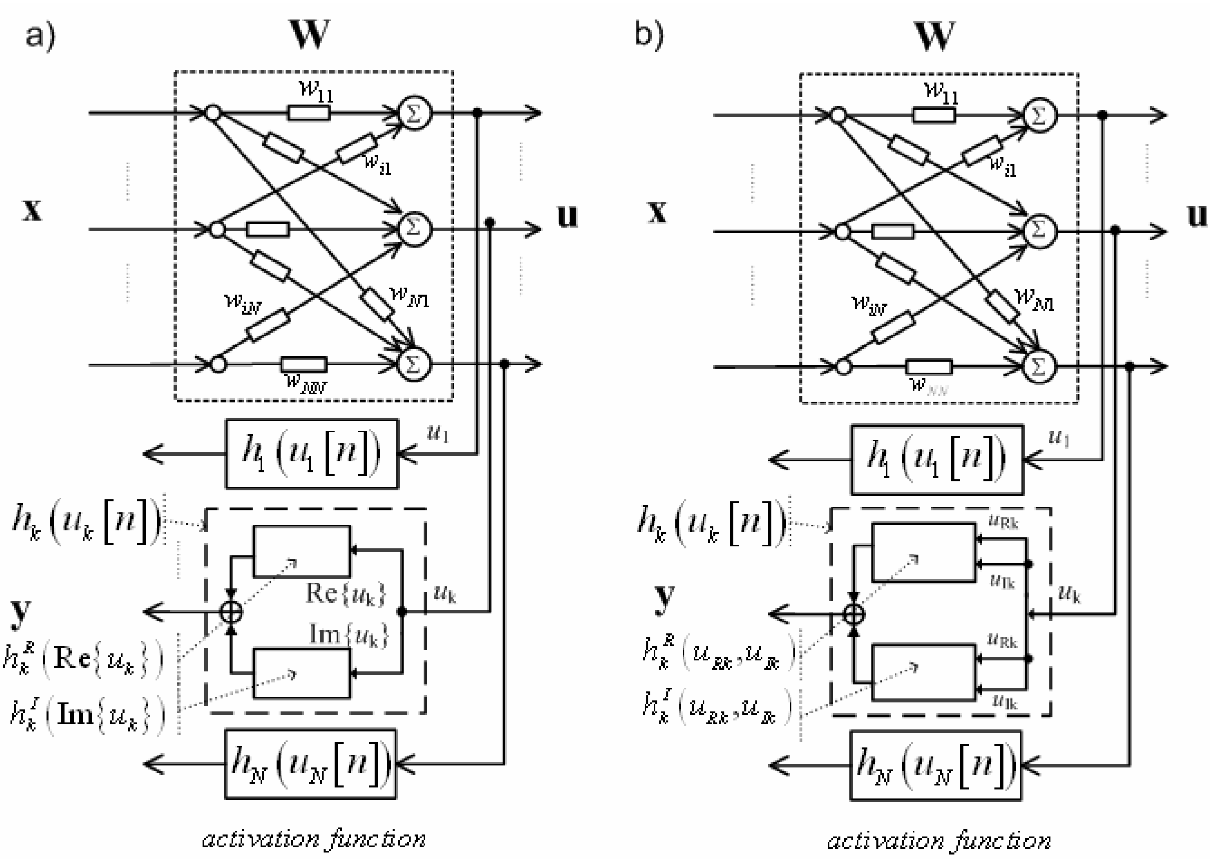


Figure 10 De-mixing model in linear environment. The activation functions are realized with splitting functions a) and generalized splitting functions b)

The Case of Nonlinear Environment

The architecture used to realize the model $\mathcal{G}\{\mathbf{G},\mathbf{W}\}$ in eq. (9) is shown in Figure 11.

In the same way as the case of linear environment the de-mixing algorithm is based on an extension of the *INFOMAX algorithm* (Bell & Sejnowski, 1995), performing the ME approach.

The network output $\mathbf{y}$ is similar to eq. (32). The network parameters are $\boldsymbol{\Phi} = \{w_{ij}, \boldsymbol{\Phi}_h, \boldsymbol{\Phi}_G \;|\forall i, j\;\}$, where w_{ij} are the entries in matrix $\mathbf{W}$, $\boldsymbol{\Phi}_h = \{Q_R^h,\ Q_I^h\}$ and $\boldsymbol{\Phi}_G = \{Q_R^G,\ Q_I^G\}$ are the spline control points for the real and imaginary part of the nonlinear compensating functions $\mathbf{G}(\bullet)$. So using eq. (33) the cost function $\mathcal{L}\{\mathbf{y}[n],\boldsymbol{\Phi}\}$ to be maximized is the joint entropy in eq. (34).

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

$$\det\left(\tilde{\mathbf{J}}\right) = \prod_{k=1}^{N} \dot{\tilde{y}}_k \dot{\tilde{r}}_k \det\left(\tilde{\mathbf{W}}\right) \tag{41}$$

In eq. (41) $\dot{\tilde{y}}_k$ and $\dot{\tilde{r}}_k$ are the derivative of the *k*-th element of $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{r}}$ respectively.

Inserting eq. (42) in the derivation of the cost function eq. (35) with respect to the elements of parameters $\boldsymbol{\Phi}$ we obtain

$$\frac{\partial}{\partial \boldsymbol{\Phi}} L\{\mathbf{y},\boldsymbol{\Phi}\} = \frac{\partial}{\partial \boldsymbol{\Phi}}\left[\ln\left(\det\left(\tilde{\mathbf{W}}\right)\right) + \sum_{k=1}^{N} \ln \dot{y}_{Rk} + \sum_{k=1}^{N} \ln \dot{y}_{Ik} + \sum_{k=1}^{N} \ln \dot{r}_{Rk} + \sum_{k=1}^{N} \ln \dot{r}_{Ik}\right] \tag{42}$$

In eq. (42) expected values have been replaced by instantaneous values.

Maximization of eq. (42) by the stochastic gradient method eq. (29) yields to five learning rules (see Vigliano et al. (2008) for major details). The learning rule for the network's weights is:

$$\Delta\mathbf{W} = \Delta\mathbf{W}_R + j\Delta\mathbf{W}_I = \mathbf{W}^{-H} + \boldsymbol{\Psi}\mathbf{r}^H \tag{43}$$

that is formally identical to eq. (37) and $\boldsymbol{\Psi}$ is defined analogously. The learning rules for the spline activation functions are:

$$\begin{aligned}\Delta\mathbf{Q}^h_{\mathrm{R},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{R})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{R})\mathbf{M}\mathbf{Q}^h_{\mathrm{R},k,i+m}}\\ \Delta\mathbf{Q}^h_{\mathrm{I},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{I})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{I})\mathbf{M}\mathbf{Q}^h_{\mathrm{I},k,i+m}}\end{aligned} \tag{44}$$

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:

$$\begin{aligned}\Delta\mathbf{Q}^G_{\mathrm{R},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{R})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{R})\mathbf{M}\mathbf{Q}^G_{\mathrm{R},k,i+m}} + \\ &\quad + \mathrm{Re}\left[\boldsymbol{\Psi}\left(\mathbf{W}^H\right)_k\right]\frac{1}{2}\mathbf{T}_\nu(u_\mathrm{R})(\mathbf{M})_m\\ \Delta\mathbf{Q}^G_{\mathrm{I},k,i+m} &= \frac{\dot{\mathbf{T}}_\nu(u_\mathrm{I})(\mathbf{M})_m}{\dot{\mathbf{T}}_\nu(u_\mathrm{I})\mathbf{M}\mathbf{Q}^G_{\mathrm{I},k,i+m}} + \\ &\quad + \mathrm{Im}\left[\boldsymbol{\Psi}\left(\mathbf{W}^H\right)_k\right]\frac{1}{2}\mathbf{T}_\nu(u_\mathrm{I})(\mathbf{M})_m\end{aligned} \tag{45}$$

where $(\mathbf{W})_k$ is a vector composed by the *m*-th column of the matrix $\mathbf{W}$. For a complete derivation of the learning rules see Vigliano et al. (2008).

An alternative algorithm can be obtained using the MMI approach deducing similar learning rules. This approach is presented in Vigliano et al. (2006b).

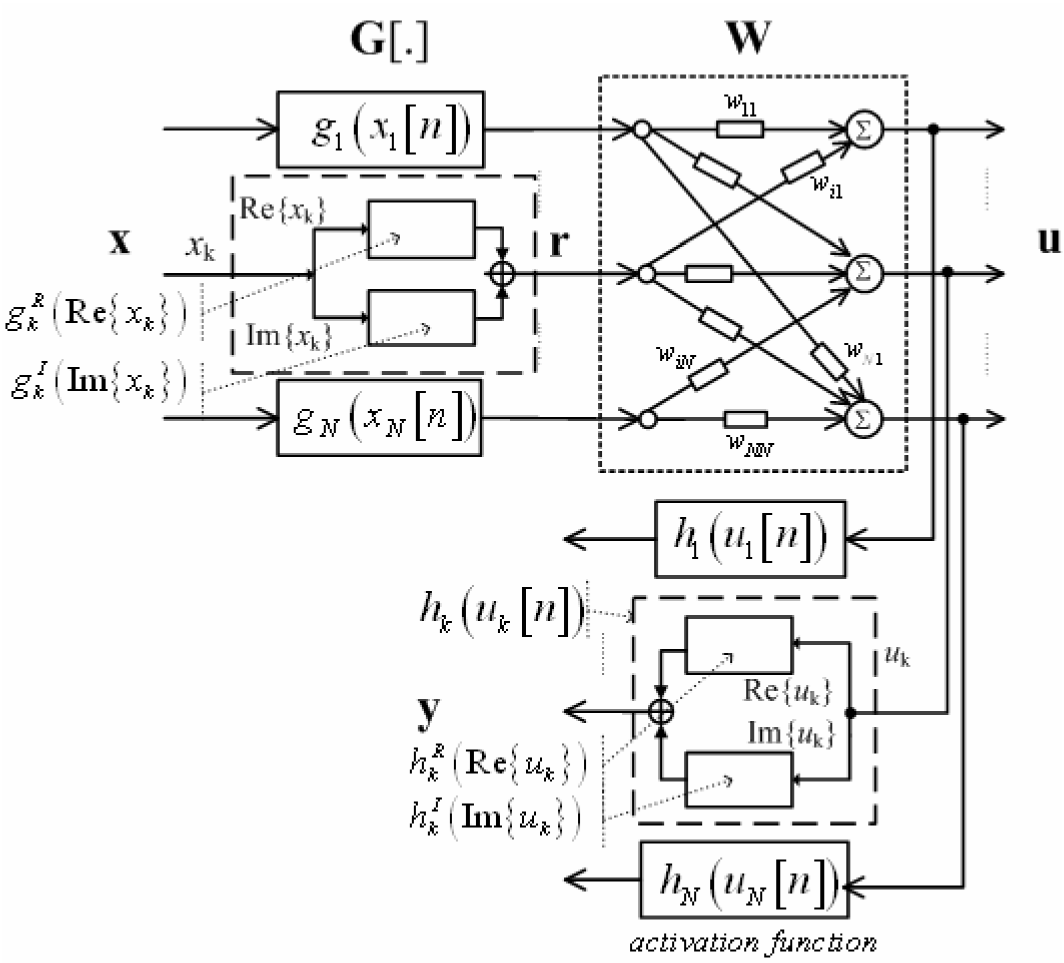


Figure 11 De-mixing model in nonlinear environment

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 was proposed (Bingham.& Hyvärinen, 2000). This algorithm is based on a non-Gaussianity maximization derived from the Negentropy (Cover & Tomas, 2006) but it works well only for circular sources.

Recently in Li & Adali (2006) a class of complex-valued ICA algorithms by maximizing the kurtosis cost function is derived.

Moreover Cardoso & Adali (2006) proposed an approach based on the maximization of the lo-likelihood function extended in Li & Adali (2008). This approach is strongly linked with the INFOMAX principle (Lee et al., 2000). A maximum likelihood (ML) solution to BSS problem was first derived in Pham et al. (1992).

Lately Novey & Adali (2008) solve the limit on circularity in the complex Fast ICA algorithm using complex analytic functions by introducing the complex maximization of non-Gaussianity (CMN) algorithm. The authors also show the connection among ICA methods through maximization of non-Gaussianity, mutual information and maximum likelihood (ML) for the complex case.

RESULTS

This section collects some experimental results in order to demonstrate the effectiveness of our complex domain approach both in linear and nonlinear environment.

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 by Schobben et al. (1999), 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

$$SIR(k) = 10\log\left[E\left\{ \left(|u|_{\sigma(k),k} \right)^2 \right\} \Big/ E\left\{ \sum_{i \neq k} \left(|u|_{\sigma(k),i} \right)^2 \right\} \right] \qquad (46)$$

In eq. (46) $u_{i,k}$ is the *i*-th output signal when only the *k*-th input signal 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.

Another way to evaluate the performance of an algorithm in linear case only is to analyze the matrix product $\mathbf{WA}$ 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 (Amari et al., 1996)

$$S = \sum_{i=1}^{N} \left(\frac{\sum_{k=1}^{N} |q_{ik}|^2}{\max_p \left[|q_{ip}|^2 \right]} - 1 \right) + \sum_{k=1}^{N} \left(\frac{\sum_{i=1}^{N} |q_{ik}|^2}{\max_p \left[|q_{pk}|^2 \right]} - 1 \right) \qquad (47)$$

The index in eq. (47) is a non-negative number and it is equal to zero only for perfect separation.

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. (19), 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.

In the first experiment we adopt the algorithm with the bi-dimensional AF in eq. (22), 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.

The mixing environment (eq. (6)) 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 1:

Table 1 Free parameters in test 1

$N_{\mathbf{h}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	n_R
21	$2{,}3\cdot10^{-3}$	$5\cdot10^{-4}$	100

In Figure 12 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 13 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 SNR 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 Adali et al. (2004) (see Figure 14); 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.

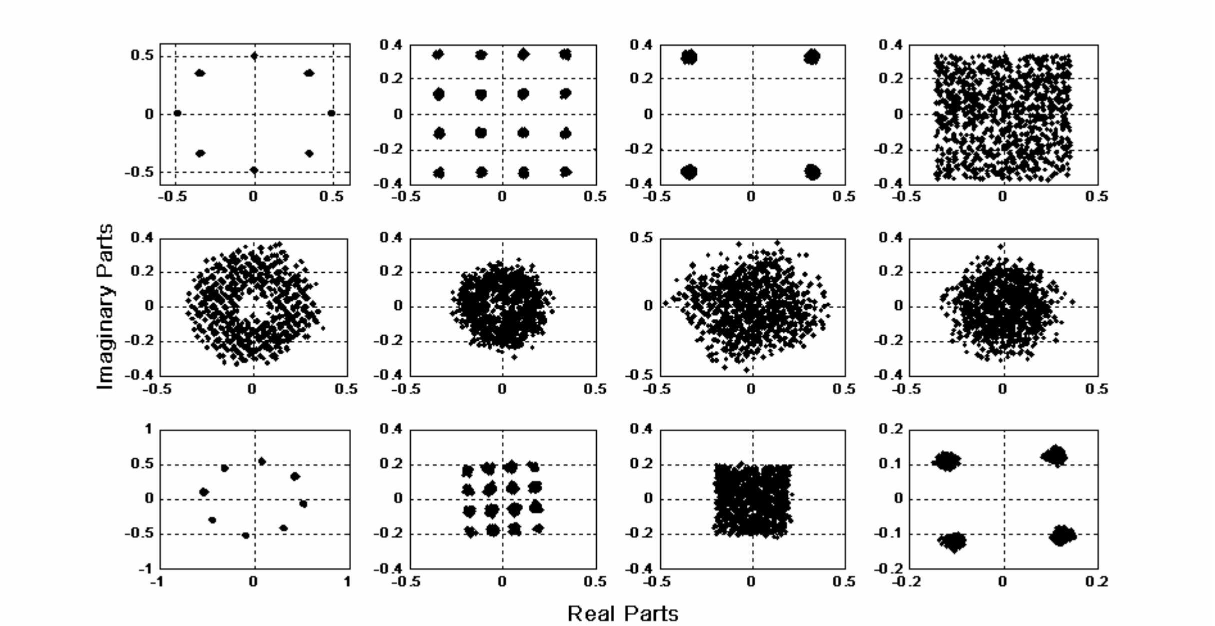


Figure 12 Scatter plot of original signals (first row), mixtures (second row) and separated signals (third row)

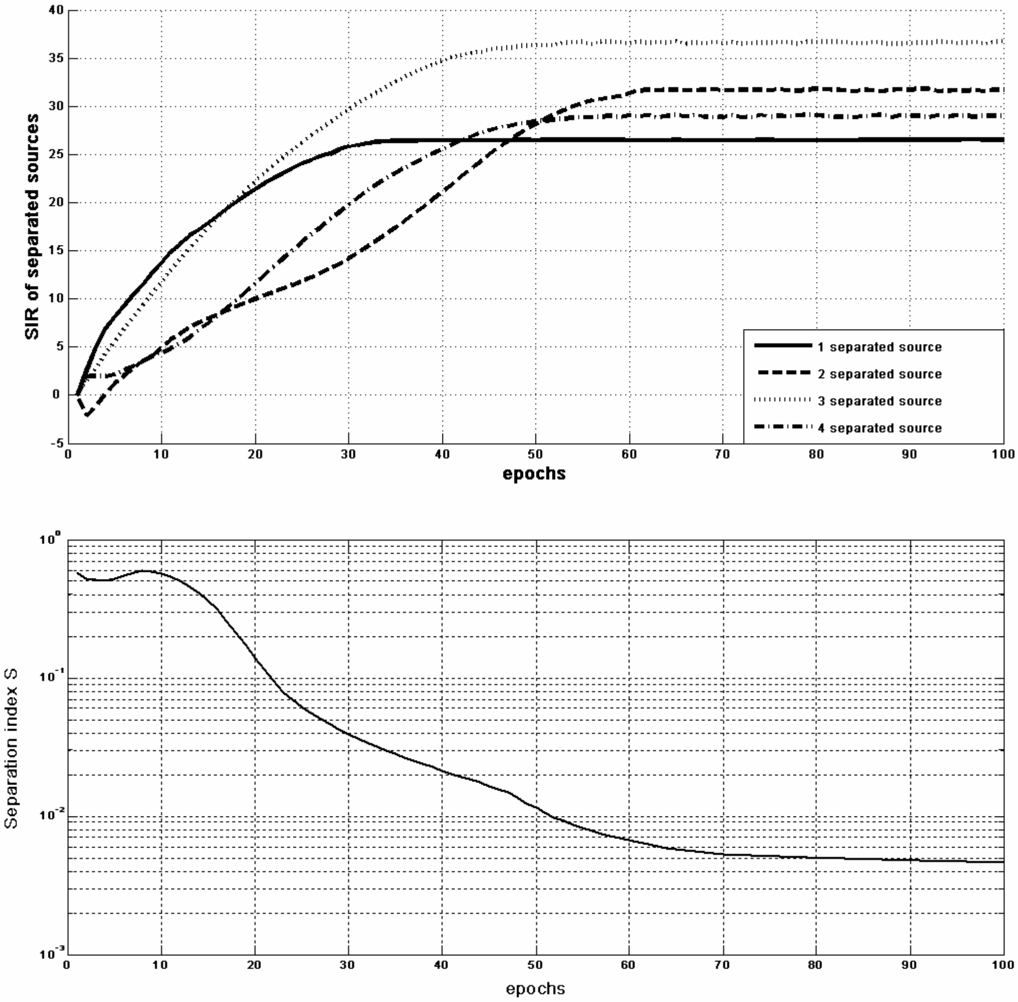


Figure 13 Performance indexes for the first experiment

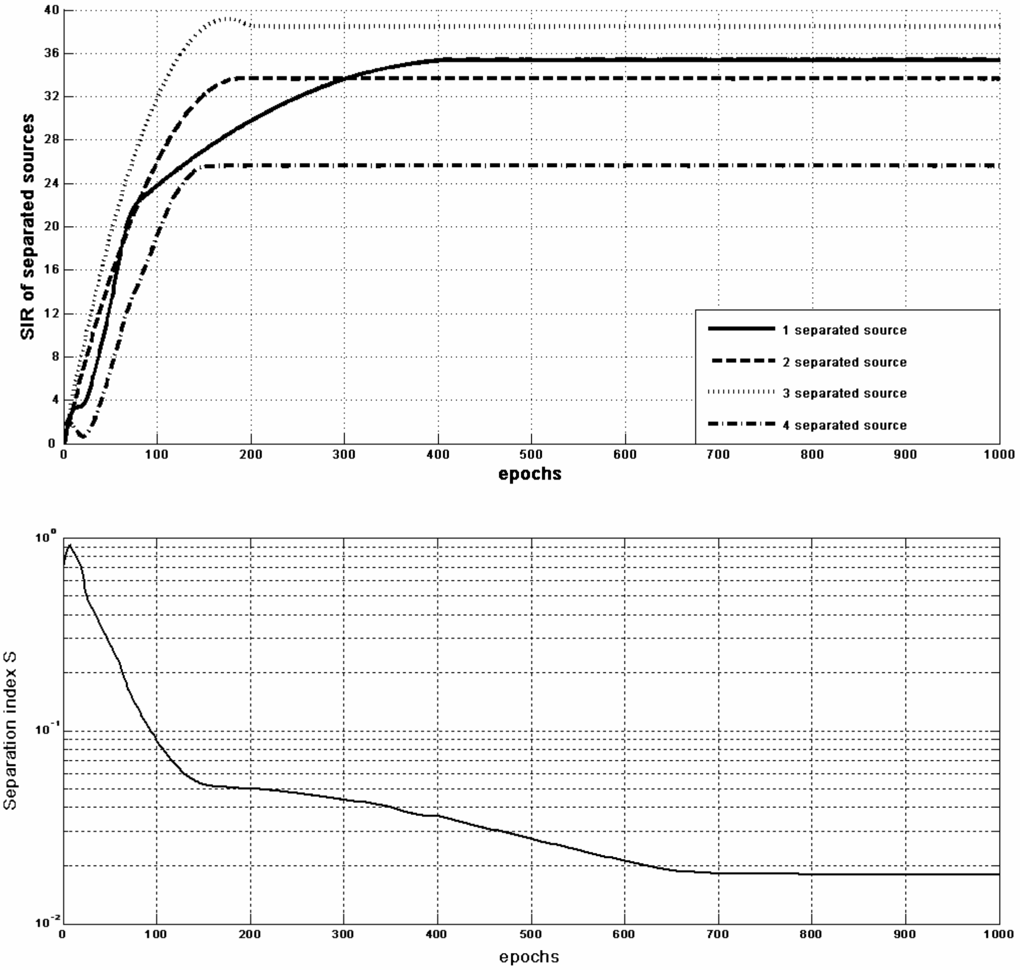


Figure 14 Performance indexes in the case of *tanh(z)* activation function

In the second experiment we adopt the algorithm with the bi-dimensional AF in eq. (22) 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. Then we modified these signals in order to correlate the real and imaginary part. 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 generic signal s_k is obtained as the mean over M past samples:

$$s_k\left(n\right) = \frac{1}{M}\sum_{p=0}^{M-1} s_k\left(n-p\right) \tag{48}$$

The mixing environment (eq. (6)) 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 2:

Table 2 Free parameters in test 2

$N_{\mathbf{h}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	n_R
21	$2{,}3\cdot 10^{-3}$	$5\cdot 10^{-4}$	100

In Figure 15 we report the performance index in eq. (47) of the algorithm versus the M parameter in eq. (48) comparing this result with respect the same test done with the algorithm described in Adali et al. (2004) 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.

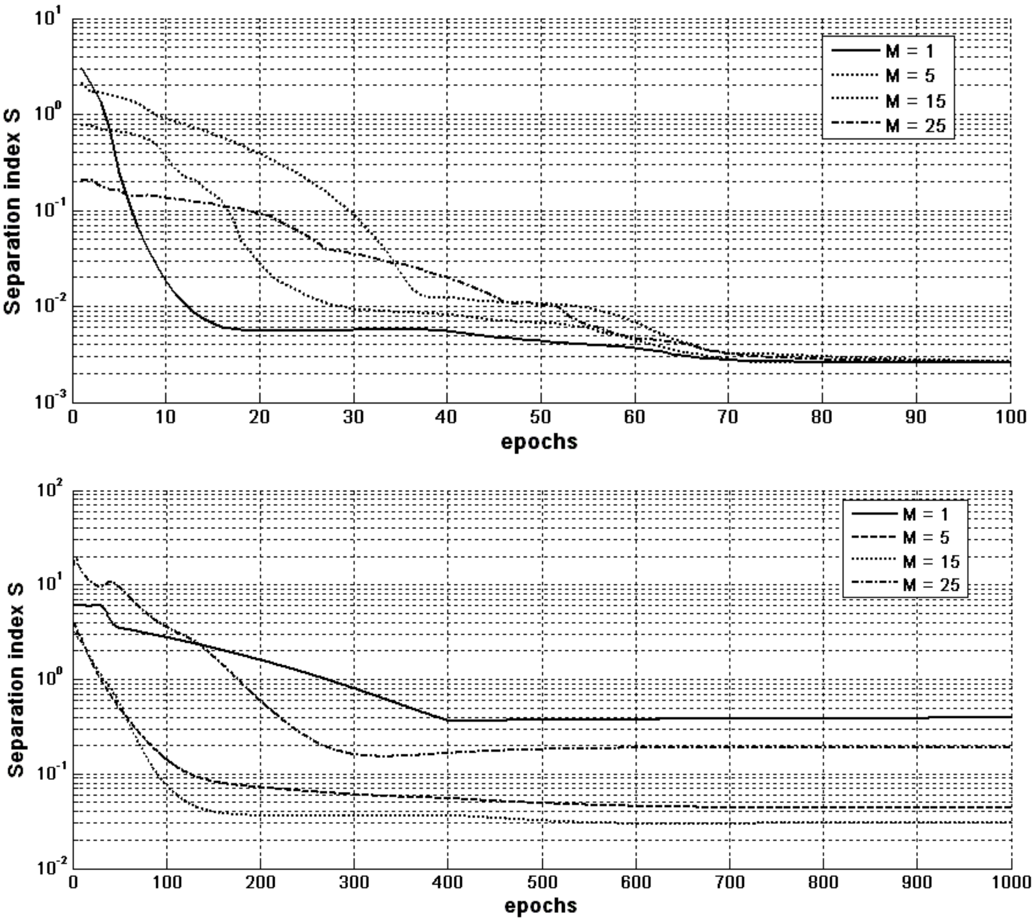


Figure 15 Performance index vs. the length *M* of the MA filter: the flexible generalized splitting activation function (top) and a *tanh(z)* activation function (bottom)

A third test is performed adopting the three signals involved in the previous example but varying the number N_{h} of spline control points in the following set of values: $\{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$. The other parameters are the same that in Table 2.

Figure 16 shows the separation index in eq. (47) for the five different values of N_{h}. The index profile shows that $N_{\text{h}} = 21$ points 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 behaviour.

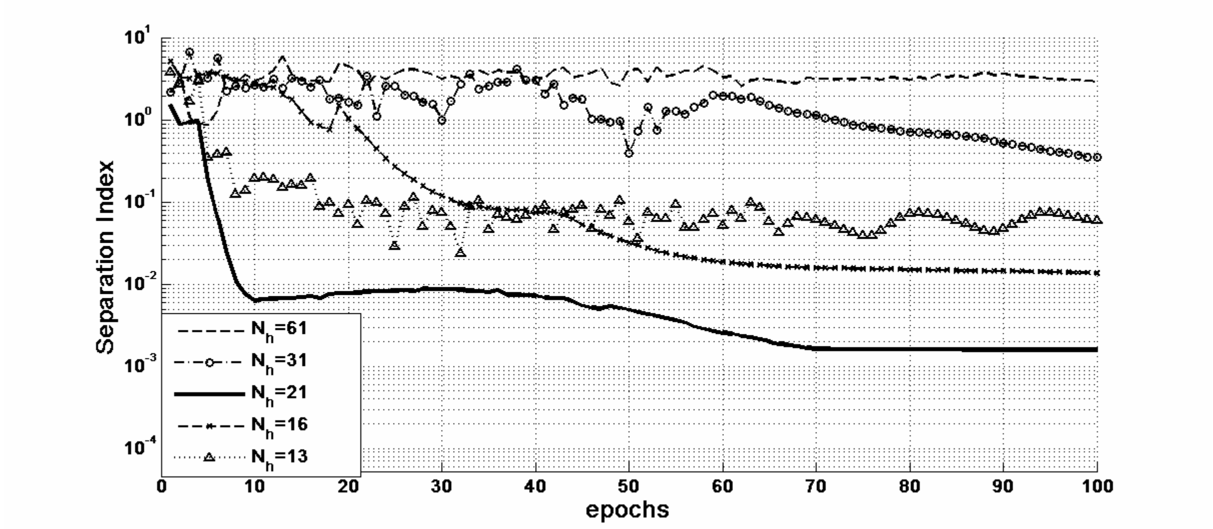


Figure 16 Performance index vs. the number of spline control points

Performance Test in PNL 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.

For the first test we adopt the algorithm with the mono-dimensional AF in eq. (20) using a 4-QAM signal, a uniform random signal and a PSK signal.

The mixing environment (eq. (8)) is:

$$\mathbf{A} = \begin{bmatrix} 0.90 - j0.30 & -0.24 + j0.36 & 0.30 + j0.06 \\ 0.30 + j0.60 & 0.24 + j0.12 & -0.06 + j0.18 \\ 0.12 - j0.24 & -0.36 + j0.60 & 0.78 + j0.36 \end{bmatrix}$$

$$\mathbf{F}[\mathbf{v}] = \begin{bmatrix} f_1[v_1] = \left(v_{R1} + 0.7v_{R1}^3\right) + j\left(v_{I1} + 0.7v_{I1}^3\right) \\ f_2[v_2] = \left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) + j\left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) \\ f_3[v_3] = \left(v_{R3} + 0.7v_{R3}^3\right) + j\left(v_{R2} + 0.7\tanh\left(3v_{R2}^3\right)\right) \end{bmatrix}$$

while the free parameters are summarized in the following Table 3:

Table 3 Free parameters in test 1

$N_{\mathbf{h}}$	$N_{\mathbf{G}}$	$\eta_{\mathbf{W}}$	$\eta_{\mathbf{h}}$	$\eta_{\mathbf{G}}$	n_R
31	31	$5\cdot10^{-5}$	$5\cdot10^{-6}$	$5\cdot10^{-6}$	400

The effectiveness of the separation is evidenced in Figure 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 18 shows that after about 200 epochs the training became stable and more accurate. So the profiles of the separation index $SIR(k)$ for each channel assures the effectiveness of the learning.

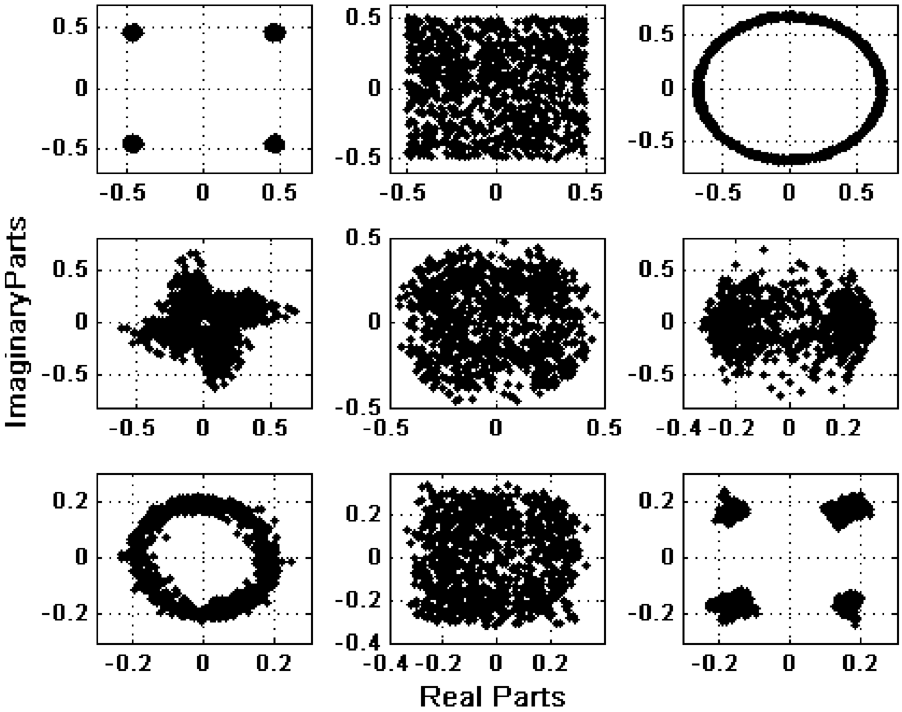


Figure 17 Scatter plot of original sources (first row), mixtures (second row) and separated sources (third row)

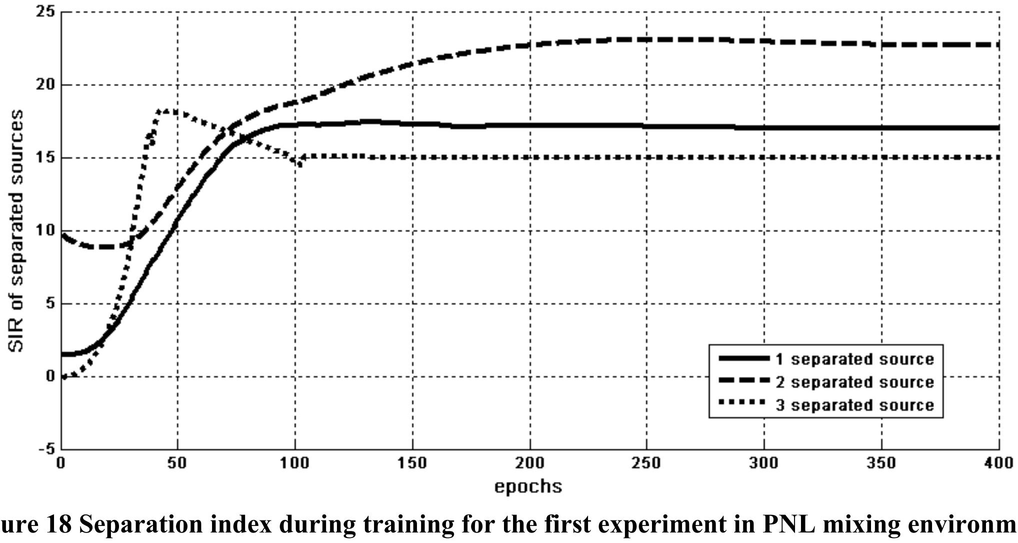


Figure 18 Separation index during training for the first experiment in PNL mixing environment

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 real and imaginary part strongly correlated.

The mixing environment (eq. (8)) and the free parameters are the same values of the previous test, see Table 3.

The effectiveness of the separation is evidenced in Figure 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 20 shows that after about 200 epochs the training became stable and more accurate. So the profiles of the separation index $SIR(k)$ for each channel assures the effectiveness of the learning.

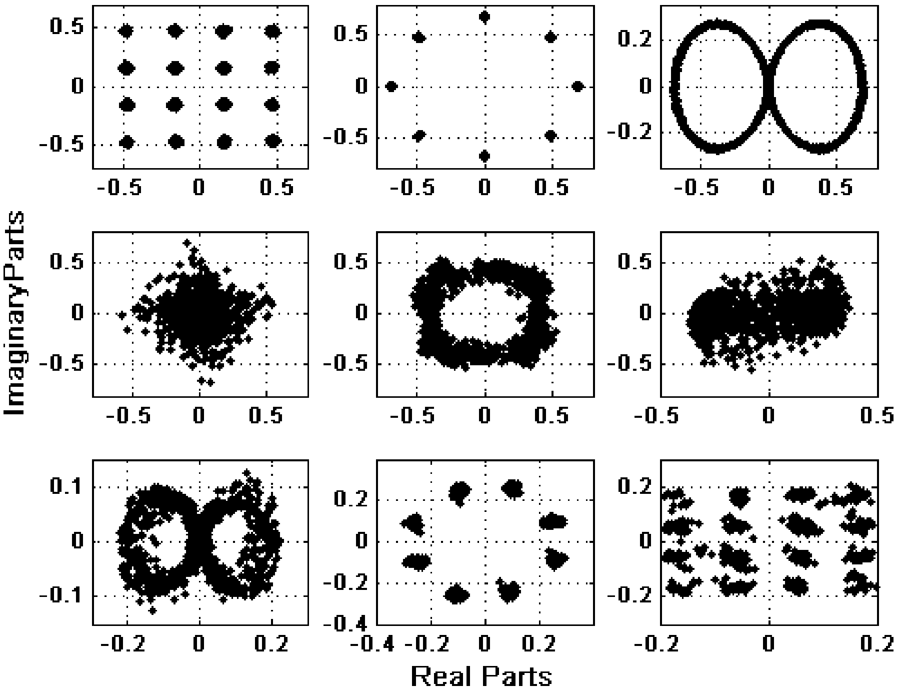


Figure 19 Scatter plot of original sources (first row), mixtures (second row) and separated sources (third row)

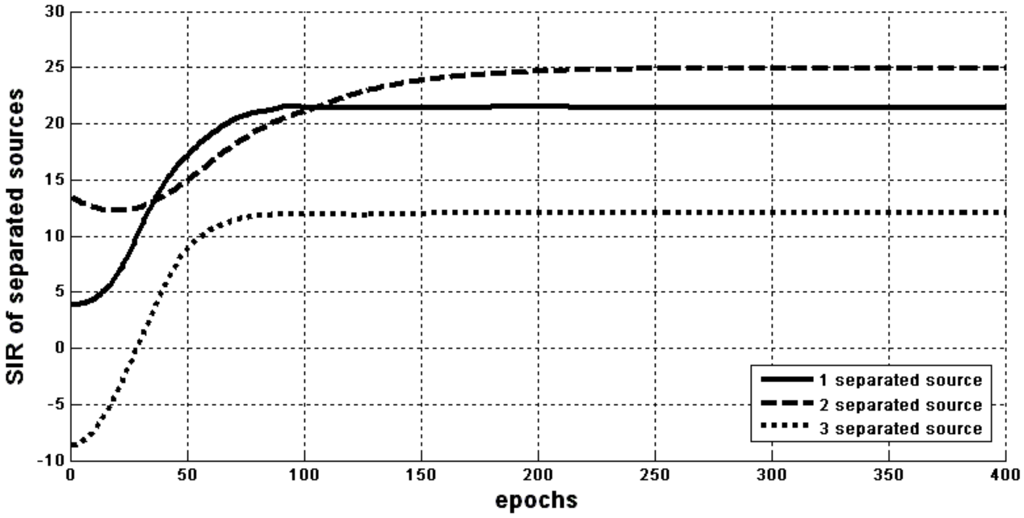


Figure 20 Separation index during training for the second experiment in PNL mixing environment

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 working on linear mixtures. The algorithm chosen for the comparison is that described in a previous section and proposed by Adali et al. (2004) 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 has been evaluated the Separation Index 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:

$$\mathbf{F}\left[\mathbf{v},\alpha,\beta,\gamma\right]=\begin{bmatrix} f_1\left[v_1,\alpha\right]=\left(v_{R1}+\alpha v_{R1}^3\right)+j\left(v_{I1}+\alpha v_{I1}^3\right) \\ f_2\left[v_2,\beta\right]=\left(v_{R2}+\beta\tanh\left(3v_{R2}^3\right)\right)+j\left(v_{R2}+\beta\tanh\left(3v_{R2}^3\right)\right) \\ f_3\left[v_3,\gamma\right]=\left(v_{R3}+\gamma v_{R3}^3\right)+j\left(v_{R2}+\gamma\tanh\left(3v_{R2}^3\right)\right) \end{bmatrix} \qquad (49)$$

The following table collects the Separation Index in eq. (46) after 600 epochs with the same learning rate of test 1 but random starting conditions with different value of $\left[\alpha,\beta,\gamma\right]$ in eq. (49) such that $\left[\alpha,\beta,\gamma\right]=\left\{\left[0,0,0\right]\quad\left[0.4,0.4,0.4\right]\quad\left[0.7,0.7,0.7\right]\quad\left[1,1,1\right]\right\}$.

Table 4: Comparison of SIR values for two algorithms

SIR(k)	$\alpha=\beta=\gamma=0$	$\alpha=\beta=\gamma=0.4$	$\alpha=\beta=\gamma=0.7$	$\alpha=\beta=\gamma=1$
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, 1.4, -7.72]	[-6.37, -6.58, -9.52]

The results collected in Table 4 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 guarantees still good results.

CONCLUSION

Considering the evolution of ICA algorithm in resolving BSS problems it is important to underline that, although in real domain environment there was been several studies, in case of complex domain mixing environments the state of art it is not so advanced.

This chapter collects a first trial to enhance the state of art of mixing environment for which ICA algorithms can provide a solution.

In this chapter 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 to the complex domain the well-known INFOMAX algorithm and is based on the use of flexible spline networks to perform local on-line estimation of the activation functions and nonlinear compensating functions.

Quality of the separation has been evaluated in terms of separation index in a number of experimental tests.

FUTURE RESEARCH DIRECTION

This is an initial work on efficient and flexible neural architectures for Blind Signal Separation in complex environment.

Necessary extensions to the research in BSS of 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.

APPENDIX

We report the proof of proposition 1.

Sufficient condition: existence of the solution.

Given the channel model in eq. (8), it is easy to verify that if **s**[*n*] is a statistically independent complex random vector, under the given assumptions, **y**[*n*] will be statistically independent too because the channel does not produce any mixing. Given the mixing model $\mathcal{F}\left\{\mathbf{A},\mathbf{F}\right\}$, assumptions a), b), and c) guarantee that there exists a matrix **W** and N functions $g_i\left(\bullet\right)$ such that:

$$\begin{aligned}&\mathbf{G}\left(\mathbf{F}\left(\mathbf{As}\left[n\right]\right)\right)=\mathbf{As}\left[n\right]\\&\mathbf{WAs}\left[n\right]=\mathbf{P\Lambda s}\left[n\right]\end{aligned}\qquad(50)$$

Based on (50), the input-output transformation can be written as $\mathbf{WG}\left[\mathbf{F}\left\{\mathbf{As}\left[n\right]\right\}\right]=\mathbf{P\Lambda s}\left[n\right]$ □

Necessary condition: uniqueness of the solution.

This condition proves that if **y**[*n*] is a statistically independent random vector, the channel model must be (8). The complex transformation which maps **s** into **y** is:

$$\begin{aligned}&\mathbf{y}\left[n\right]=\mathbf{WG}\left[\mathbf{F}\left\{\mathbf{As}\left[n\right]\right\}\right]=\\&=\mathbf{W}\begin{bmatrix}g_{R1}\left(f_{R1}\left(\sum_{i=1}^{N}a_{1i}s_i\left[n\right]\right)\right)+jg_{I1}\left(f_{I1}\left(\sum_{i=1}^{N}a_{1i}s_i\left[n\right]\right)\right)\\\vdots\\g_{RN}\left(f_{RN}\left(\sum_{i=1}^{N}a_{Ni}s_i\left[n\right]\right)\right)+jg_{IN}\left(f_{IN}\left(\sum_{i=1}^{N}a_{Ni}s_i\left[n\right]\right)\right)\end{bmatrix}\end{aligned}\qquad(51)$$

It is possible to rewrite the complex map (51) in a real form as follows:

$$\begin{aligned}&\tilde{\mathbf{y}}=\begin{bmatrix}\mathbf{y}_R\left[n\right]\\\mathbf{y}_I\left[n\right]\end{bmatrix}=\begin{bmatrix}\mathbf{W}_R&-\mathbf{W}_I\\\mathbf{W}_I&\mathbf{W}_R\end{bmatrix}\begin{bmatrix}\mathbf{G}_R\left\{\mathbf{x}_R\left[n\right]\right\}\\\mathbf{G}_I\left\{\mathbf{x}_I\left[n\right]\right\}\end{bmatrix}=\tilde{\mathbf{W}}\tilde{\mathbf{r}}\\&\begin{bmatrix}\mathbf{x}_R\left[n\right]\\\mathbf{x}_I\left[n\right]\end{bmatrix}=\begin{bmatrix}\mathbf{F}_R\left\{\mathbf{v}_R\left[n\right]\right\}\\\mathbf{F}_I\left\{\mathbf{v}_I\left[n\right]\right\}\end{bmatrix}\\&\tilde{\mathbf{v}}\left[n\right]=\begin{bmatrix}\mathbf{v}_R\left[n\right]\\\mathbf{v}_I\left[n\right]\end{bmatrix}=\begin{bmatrix}\mathbf{A}_R&-\mathbf{A}_I\\\mathbf{A}_I&\mathbf{A}_R\end{bmatrix}\begin{bmatrix}\mathbf{s}_R\left[n\right]\\\mathbf{s}_I\left[n\right]\end{bmatrix}=\tilde{\mathbf{A}}\tilde{\mathbf{s}}\end{aligned}\qquad(52)$$

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 $2N\times2N$ matrices. It is important to underline that in (52) there are only real elements. For assumption (a) matrix **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 $\tilde{\mathbf{s}}$ can be written as a function of the pdf of $\tilde{\mathbf{y}}$:

$$p_{\tilde{\mathbf{s}}}\left(\tilde{\mathbf{s}}\right)=\prod_{i=1}^{N} p_{s_{Ri}}\left(s_{Ri}\right)p_{s_{Ii}}\left(s_{Ii}\right)=$$
$$=\prod_{i=1}^{N} p_{Ry_i}\left(\begin{array}{l}\sum_{j=1}^{N} w_{Rij}g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm}-\sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\ -\sum_{j=1}^{N} w_{Iij}g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm}+\sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right]\end{array}\right)p_{y_{Ii}}\left(y_{Ii}\right)\left|\tilde{\mathbf{J}}\right| \qquad \forall\tilde{\mathbf{s}}\in\mathbb{R}^{2N} \tag{53}$$

in which $\tilde{\mathbf{J}}$ is the Jacobian matrix of the application which maps $\tilde{\mathbf{s}}$ into $\tilde{\mathbf{y}}$.

From assumption (d) $\exists\overline{\mathbf{s}}\in\mathbb{C}^N$ such that $p_{\mathbf{s}}\left(\overline{\mathbf{s}}\right)\equiv 0$. Then considering $\tilde{\mathbf{s}}[n]=\left[\mathbf{s}_R[n],\mathbf{s}_I[n]\right]^T$ in which $\mathbf{s}_R[n],\mathbf{s}_I[n]\in\mathbb{R}$ the assumption (d) can be reformulated as follows: $\exists\overline{\tilde{\mathbf{s}}}\in\mathbb{R}^{2N}$ $p_{\tilde{\mathbf{s}}}\left(\overline{\tilde{\mathbf{s}}}\right)$=0 .

From (51), for a non null Jacobian $\tilde{\mathbf{J}}$, there exists some $\tilde{\mathbf{y}}^0=\left[y_{R1}^0,\cdots,y_{RN}^0,y_{I1}^0,\cdots y_{IN}^0\right]\in\mathbb{R}^{2N}$ such that $\prod_{i=1}^{N} p_{y_{Ri}}\left(y_{Ri}^0\right)p_{y_{Ii}}\left(y_{Ii}^0\right)=0$. Consequently there exists at least one integer i such that $p_{y_{Ri}}\left(y_{Ri}^0\right)=0$ or $p_{y_{Ii}}\left(y_{Ii}^0\right)=0$. This leads to the following equation:

$$\begin{array}{l}\tilde{y}_i^0=\sum_{j=1}^{N} w_{Rij}g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm}-\sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\ -\sum_{j=1}^{N} w_{Iij}g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Rjm}s_{Im}+\sum_{m=1}^{N} a_{Ijm}s_{Rm}\right)\right]\end{array} \tag{54}$$

in which $\tilde{y}_i^0$ is the i^{th} element of the vector $\tilde{\mathbf{y}}^0$.

Solutions of (54) lie on $\mathcal{H}_i\left(\tilde{\mathbf{s}}\right)$, which is a hypersurface in $\mathbb{R}^{2N}$. It is evident that $\forall\tilde{\mathbf{s}}\in\mathcal{H}_i\left(\tilde{\mathbf{s}}\right)\Rightarrow p_{\tilde{\mathbf{s}}}\left(\tilde{\mathbf{s}}\right)=0$. For a given *i*, $\mathcal{H}_i\left(\tilde{\mathbf{s}}\right)$ 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 $\mathcal{H}_i\left(\tilde{\mathbf{s}}\right)$ is not parallel to any $\tilde{s}_i=0$ plane. The projection of $\mathcal{H}_i\left(\tilde{\mathbf{s}}\right)$ onto $\tilde{s}_i$ should be $\mathbb{R}$: $\forall\tilde{s}_i\in\mathbb{R}\ \exists\,\tilde{s}_1,\ldots,\tilde{s}_{i\text{-}1},\tilde{s}_{i+1},\ldots,\tilde{s}_N,\ldots,\tilde{s}_{2N}:\tilde{\mathbf{s}}\in\mathcal{H}_i\Rightarrow p_{\tilde{\mathbf{s}}}\left(\tilde{\mathbf{s}}\right)\equiv 0$. This cannot be true since $\int_S p_{\tilde{\mathbf{s}}}\left(\tilde{\mathbf{s}}\right)d\tilde{s}=1$. Without loss of generality, it can be noted that:

$$\begin{array}{l}\sum_{j=1}^{N} w_{Rij}g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm}-\sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\ -\sum_{j=1}^{N} w_{Iij}g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm}+\sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right]=b_{\sigma(i)}\left(\tilde{s}_{\sigma(i)}\right) \quad i=1,\ldots,N\\ \sum_{j=1}^{N} w_{Iij}g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm}-\sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\ +\sum_{j=1}^{N} w_{Rij}g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm}+\sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right]=b_{\sigma(i)}\left(\tilde{s}_{\sigma(i)}\right) \quad i=N+1,\ldots,2N\end{array} \tag{55}$$

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 σ(*i*)=*i*

$$\begin{aligned}
&\sum_{j=1}^{N} w_{Rij} g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm} - \sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\
&-\sum_{j=1}^{N} w_{Iij} g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm} - \sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right] = b_i\left(\tilde{s}_i\right) \quad i=1...N\\
&\sum_{j=1}^{N} w_{Iij} g_{Rj}\left[f_{Rj}\left(\sum_{m=1}^{N} a_{Rjm}s_{Rm} - \sum_{m=1}^{N} a_{Ijm}s_{Im}\right)\right]+\\
&+\sum_{j=1}^{N} w_{Rij} g_{Ij}\left[f_{Ij}\left(\sum_{m=1}^{N} a_{Ijm}s_{Rm} - \sum_{m=1}^{N} a_{Rjm}s_{Im}\right)\right] = b_i\left(\tilde{s}_i\right) \quad i=N+1...2N
\end{aligned} \qquad (56)$$

Derivation with respect to **s** yields

$$\begin{bmatrix} \dot{b}_1\left(s_1\right) & 0 \\ 0 & \dot{b}_{2N}\left(\tilde{s}_{2N}\right) \end{bmatrix} = \tilde{\mathbf{W}} \begin{bmatrix} \dot{g}_{R1}\left[f_{R1}\left(\mathbf{v}_R\left(\tilde{\mathbf{s}}\right)\right)\right] & 0 \\ 0 & \dot{g}_{IN}\left[\left(\mathbf{x}_I\left(\tilde{\mathbf{s}}\right)\right)\right] \end{bmatrix}. \begin{bmatrix} \dot{f}_{R1}\left(\mathbf{v}_R\left(\tilde{\mathbf{s}}\right)\right) & 0 \\ 0 & \dot{f}_{IN}\left(\mathbf{v}_I\left(\tilde{\mathbf{s}}\right)\right) \end{bmatrix} \tilde{\mathbf{A}} \qquad (57)$$

Considering $\tilde{\mathbf{s}}_1$ and $\tilde{\mathbf{s}}_2$ as coordinates of the hypersurface $\mathcal{H}\left(\tilde{\mathbf{s}}\right)$, (57) can be evaluated in $\tilde{\mathbf{s}}_1$ and in $\tilde{\mathbf{s}}_2$ as follows:

$$\begin{cases} \mathbf{D}\left(\tilde{\mathbf{s}}_1\right) = \tilde{\mathbf{W}}\mathbf{\Lambda}_{\dot{\mathbf{G}}}\left(\tilde{\mathbf{s}}_1\right)\mathbf{\Lambda}_{\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_1\right)\tilde{\mathbf{A}} \\ \mathbf{D}\left(\tilde{\mathbf{s}}_2\right) = \tilde{\mathbf{W}}\mathbf{\Lambda}_{\dot{\mathbf{G}}}\left(\tilde{\mathbf{s}}_2\right)\mathbf{\Lambda}_{\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_2\right)\tilde{\mathbf{A}} \end{cases} \rightarrow \begin{cases} \mathbf{D}\left(\mathbf{s}_1\right) = \tilde{\mathbf{W}}\mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_1\right)\tilde{\mathbf{A}} \\ \mathbf{D}\left(\mathbf{s}_2\right) = \tilde{\mathbf{W}}\mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_2\right)\tilde{\mathbf{A}} \end{cases} \qquad (58)$$

Then eliminating $\tilde{\mathbf{W}}$:

$$\tilde{\mathbf{A}} \quad \underbrace{\mathbf{D}^{-1}\left(\tilde{\mathbf{s}}_2\right)\mathbf{D}\left(\tilde{\mathbf{s}}_1\right)}_{\begin{bmatrix} d_{11}(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1) & & 0 \\ & \ddots & \\ 0 & & d_{2N2N}(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1) \end{bmatrix}} \quad = \quad \underbrace{\mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}{}^{-1}\left(\tilde{\mathbf{s}}_2\right)\mathbf{\Lambda}_{\dot{\mathbf{G}}\dot{\mathbf{F}}}\left(\tilde{\mathbf{s}}_1\right)}_{\begin{bmatrix} \lambda_{11}(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1) & & 0 \\ & \ddots & \\ 0 & & \lambda_{2N2N}(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1) \end{bmatrix}} \quad \tilde{\mathbf{A}}\,.$$

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:

$$\begin{cases} \tilde{a}_{ij}\left[d_{jj}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right)-\lambda_{ii}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right)\right]=0 \\ \tilde{a}_{hj}\left[d_{jj}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right)-\lambda_{hh}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right)\right]=0 \end{cases} \Rightarrow \quad \lambda_{ii}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right)=\lambda_{hh}\left(\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1\right) \quad \forall\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1 \in \mathcal{H} \quad (59)$$

in which $\tilde{a}_{ij}$ for *i*,*j*=1,…,2*N* is an element of $\tilde{\mathbf{A}}$. From (59) it follows:

$$\frac{\dot{\tilde{g}}_i\left[\hat{\tilde{f}}_i\left(\left(\tilde{\mathbf{A}}\right)_{\sigma(i)}\tilde{\mathbf{s}}_1\right)\right]\dot{\hat{\tilde{f}}}_i\left(\left(\tilde{\mathbf{A}}\right)_i\tilde{\mathbf{s}}_1\right)}{\dot{\tilde{g}}_h\left[\hat{\tilde{f}}_h\left(\left(\tilde{\mathbf{A}}\right)_h\tilde{\mathbf{s}}_2\right)\right]\dot{\hat{\tilde{f}}}_h\left(\left(\tilde{\mathbf{A}}\right)_h\tilde{\mathbf{s}}_2\right)} = C \quad i=1,...,2N \;\forall\tilde{\mathbf{s}}_2,\tilde{\mathbf{s}}_1 \qquad (60)$$

where $\tilde{f}_i$ is the i^{th} element of the vector $\left[\mathbf{F}_R,\mathbf{F}_I\right]^T$, $\hat{\tilde{f}}_i\left(\left(\tilde{\mathbf{A}}\right)_i\tilde{\mathbf{s}}_1\right)=\alpha\tilde{f}_i\left(\left(\tilde{\mathbf{A}}\right)_i\tilde{\mathbf{s}}_1\right)$ and *C* is a constant. For the two linear forms in (59) $\left(\tilde{\mathbf{A}}\right)_i\mathbf{s}_1$ $\left(\tilde{\mathbf{A}}\right)_h\mathbf{s}_2$ are independent as assumed in (a), it is possible to express the (59) in the following way: $\dot{\tilde{g}}_i\left[\hat{f}_i\left(x\right)\right]\dot{\hat{\tilde{f}}}_i\left(x\right)=C\dot{\tilde{g}}_h\left[\hat{\tilde{f}}_h\left(y\right)\right]\dot{\hat{\tilde{f}}}_h\left(y\right) \quad \forall x,y\in\mathbb{R}$. This can be true if and only if $\tilde{g}_i\left(\cdot\right)$ is the inverse of $\tilde{f}_i\left(\cdot\right)$, up to a scaling factor.

From (52), by previous results it follows:

$$\tilde{\mathbf{y}}[n]=\tilde{\mathbf{W}}\begin{bmatrix}\xi_1 & & \\ & \ddots & \\ & & \xi_{2N}\end{bmatrix}\tilde{\mathbf{A}}\tilde{\mathbf{s}}[n] \qquad (61)$$

where $\xi_1, \ldots, \xi_{2N}$ are scaling coefficients.

This method reduces the mixing-demixing non-linear channel (50) to the simpler, linear model (61). By considering:

$$\begin{aligned}\tilde{\mathbf{W}}' &= \tilde{\mathbf{W}}\begin{bmatrix}\xi_1 & & \\ & \ddots & \\ & & \xi_N\end{bmatrix} \\ \tilde{\mathbf{y}}[n] &= \tilde{\mathbf{W}}\begin{bmatrix}\xi_1 & & \\ & \ddots & \\ & & \xi_{2N}\end{bmatrix}\tilde{\mathbf{A}}\tilde{\mathbf{s}}[n]\end{aligned} \qquad (62)$$

(61) can be transformed into $\tilde{\mathbf{y}}=\tilde{\mathbf{W}}'\tilde{\mathbf{A}}\tilde{\mathbf{s}}$. For this formulation independent outputs can be obtained if and only if

$$\tilde{\mathbf{W}}'\tilde{\mathbf{A}}=\tilde{\mathbf{P}}\begin{bmatrix}\varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_{2N}\end{bmatrix} \qquad (63)$$

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 (63) in the following way:

$$\begin{bmatrix}\mathbf{W}_R\mathbf{D}_1 & -\mathbf{W}_I\mathbf{D}_1 \\ \mathbf{W}_I\mathbf{D}_2 & \mathbf{W}_R\mathbf{D}_2\end{bmatrix}\begin{bmatrix}\mathbf{A}_R & -\mathbf{A}_I \\ \mathbf{A}_I & \mathbf{A}_R\end{bmatrix}=\begin{bmatrix}\mathbf{P}_R & -\mathbf{P}_I \\ \mathbf{P}_I & \mathbf{P}_R\end{bmatrix}\begin{bmatrix}\varepsilon_1 & & \\ & \ddots & \\ & & \varepsilon_{2N}\end{bmatrix} \qquad (64)$$

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 $\begin{cases}p_{i,j}\neq 0 \Leftrightarrow p_{i+N,j+N}\neq 0 \\ p_{i,j+N}\neq 0 \Leftrightarrow p_{i+N,j}\neq 0\end{cases}$ $i,j\in 1,..,N$. □

REFERENCES

Adali, T. & Li, H. (2007), 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)*, pp. 633-636.

Adali, T., Kim, T. & Calhoun, V. D. (2004), Independent Component Analysis by Complex Nonlinearities, *in Proc. ICASSP 2004*, pp. 525-528.

Amari, S., Cichocki, A. & Yang, H. H. (1996), A New Learning Algorithm for Blind Signal Separation, *Advances in Neural Information Processing Systems* **8**, 757-763.

Bell, A. J. & Sejnowski, T. J. (1995), An information-maximization approach to blind separation and blind deconvolution, *Neural Computation* **7**, 1129-1159.

Benvenuto, N., Marchesi, M., Piazza, F. & Uncini, A. (1991), Nonlinear satellite radio links equalized using Blind Neural Network, *in Proc. of ICASSP*, pp. 1521-1524.

Benvenuto, N. & Piazza, F. (1992), On the complex backpropagation algorithm, *IEEE Transactions on Signal Processing* **40**(4), 967-969.

Bingham, E. & Hyvärinen, A. (2000), A fast fixed-point algorithm for independent component analysis of complex valued signals, *International Journal of Neural Systems*

10(1), pp. 1-8.

Calhoun, V. D. & Adali, T. (2002), Complex Infomax: Convergence and Approximation of Infomax With Complex Nonlinearities, *in Proc. NNSP*, pp. 307-316.

Calhoun, V. D., Adali, T., Pearlson, G. D. & Pekar, J. J. (2002a), On Complex Infomax Applied to Complex FMRI Data, *in Proc. ICASSP*, Vol. I, pp. 1009-1012.

Calhoun, V. D., Adali, T., Pearlson, G. D., Van Zijl, P. C. & Pekar, J. J. (2002b), Independent Component Analysis of FMRI Data in the Complex Domain, *Magn. Reson. Med.* **48**, 180-192.

Cardoso, J. & Laheld, B. H. (1996), Equivariant Adaptive Source Separation, *IEEE Trans. on Signal Processing* **45**(2), 3014-3030.

Cardoso, J. & Adali, T. (2006), The Maximum Likelihood Approach to Complex ICA, *in Proc. of Int. Conf. on Acoustic, Speech and Signal Processing*, pp. 673-676.

Catmull, E. & Rom, R. (1974), A class of local interpolating splines. In Barnhill, R.E. & Riesenfeld, R.F. (Eds.) *Computer-Aided Geometric Design*, pp. 317-326, New York Accademic.

Choi, S., Cichocki, A. & Amari, S. (2000), Flexible Independent Component Analysis, *Journal of VLSI Signal Processing - Systems for, Image and video Technology* **X**(1).

Cichocki, A. & Amari, S. (2002), *Adaptive Blind Signal and Image Processing*, John Wiley.

Clarke, T. L. (1990), Generalization of Neural Networks to the Complex Plane, *in Proc. of IJCNN*, pp. 435-440.

Comon, P. (1994), Independent Component Analysis, A new Concept?, *Signal Processing* **36**, 287-314.

Cover, T. M. & Thomas, J. A. (2006), *Elements of Information Theory*, John WIley & Sons, Inc., 2nd edition.

Cybenko, G. (1989), Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems* **2**, 303-314.

Eriksson, J. & Koivunen, V. (2002), Blind Identifiability of Class of Nonlinear Instantaneous ICA Models, *in XI European Signal Processing Conference EUSIPCO2002*, pp. 7-10.

Eriksson, J. & Koivunen, V. (2004), Identifiability, separability and uniqueness of linear ICA models, *Signal Processing Letters, IEEE* **11**(7), 601-604.

Eriksson, J. & Koivunen, V. (2006), Complex random vectors and ICA models: identifiability, uniqueness, and separability, *Information Theory, IEEE Transactions on* **52**(3), 1017-1029.

Fiori, S., Uncini, A. & Piazza, F. (1999), Neural Blind Separation of Complex Sources by Extended APEX Algorithm (EAPEX), *in Proc. of IEEE ISCAS'99*, pp. 627-630.

Fiori, S. (2000), Blind Signal Processing by the Adaptive Activation function Neuron, *Neural Networks* **13**(6), 597-611.

Georgiou, G. M. & Koutsougeras, C. (1992), Complex Domain Backpropagation, *IEEE Trans. On Circuit and System II* **39**(5), 330-334.

Guarnieri, S., Piazza, F. & Uncini, A. (1999), Multilayer Feedforward Networks with Adaptive Spline Activation Function, *IEEE Trans. on Neural Network* **10**(3), 672-683.

Haykin, S. (1999), *Neural Networks, a comprehensive foundation*, Prentice-Hall.

Haykin, S., (Ed.). (2000), *Unsupervised Adaptive Filtering, Volume1: Blind Source Separation*, John Wiley & Sons, Inc..

Hyvarinen, A. & Oja, E. (2000), Independent component analysis: algorithms and applications, *Neural Networks* **13**, 411-430.

Hyvärinen, A.; Karhunen, J. & Oja, E. (2001), *Independent Component Analysis*, John Wiley & Sons, Inc..

Jutten, C. & Herault, J. (1991), Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic Architecture, *Signal Processing* **24**, 1-10.

Jutten, C. & Karhunen, J. (2003), Advances in Nonlinear Blind Sources Separation, *in 4th International Symposium on Independent Component Analisys and Blind Signal Separation (ICA2003),* pp. 245-256.

Kim, T. & Adali, T. (2000), 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,* vol.1, pp. 231-240.

Kim, T. & Adali, T. (2001a), Complex backpropagation neural network using elementary trascende ntal activation functions, *in Proc. of IEEE ICASSP*, pp. 1281-1284.

Kim, T. & Adali, T. (2001b), Approximation by fully complex MLP using elementary transcendental activation functions, *in Proceedings of Neural Networks for Signal Processing XI*, pp. 203-212.

Kim, T. & Adali, T. (2002a), Fully complex multilayer perceptron for nonlinear signal processing, *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* **32**, 29-43.

Kim, T. & Adali, T. (2002b), Universal approximation of fully complex feed-foreward neural network, *in Proc. Of IEEE ICASSP'02,* pp. 973-976.

Lee, T. W., Girolami, M., Bell, A. J. & Sejnowsky, T. J. (2000), A Unifying Information-Theoretic framework for independent Component Analysis, *In Computer & Mathematics with applications,* **39**(11), pp. 11-21.

Leung, H. & Haykin, S. (1991), The complex backpropagation algorithm, *IEEE Transactions on Signal Processing* **39**(9), 2101-2104.

Li, H. & Adali, T. (2006), Gradient and Fixed-Point Complex ICA Algorithms Based on Kurtosis Maximization, *in Proceedings of the 16th Workshop on Machine Learning for Signal Processing*, pp. 85-90.

Li, H. & Adali, T. (2008), Stability analysis of complex maximum likelihood ICA using Wirtinger calculus, *in Proc. of IEEE Int. Conf. on Acoustic, Speech and Signal Processing*, pp. 1801-1804.

Novey, M. & Adali, T. (2008), Complex ICA by Negentropy Maximization, *IEEE Transaction on Neural Networks* **19**(4), 596-609.

Papoulis, A. (1991), *Probability, Random Variables and Stochastic Process*, McGraw-Hill.

Pham, D. T., Garrat, D. & Jutten, C. (1992), Separation of mixture of independent sources through maximum likelihood approach, *in Proc. EUSIPCO*, pp. 771-774.

Piazza, F., Uncini, A. & Zenobi, M. (1993), Neural Networks with Digital LUT Activation Function, *in Proceedings of IJCNN*, pp. 1401-1404.

Pierani, A., Piazza, F., Solazzi, M. & Uncini, A. (2000), Low Complexity Adaptive Non-Linear Function for Blind Signal Separation, *in Proc. of IEEE IJCNN2000*, pp. 333-338.

Roberts, S. & Everson, R., (Ed.). (2001), *Independent Component Analysis: Principles and Practice,* Cambridge University Press.

Scarpiniti, M., Vigliano, D., Parisi, R. & Uncini, A. (2007), Generalized Flexible Splitting Function Outperforms Classical Approaches in Blind Signal Separation of Complex Environment, *in Proc. of DSP2007*, pp. 215-218.

Scarpiniti, M., Vigliano, D., Parisi, R. & Uncini, A. (2008), Generalized Splitting Functions for Blind Separation of Complex Signals, *Neurocomputing,* DOI: 10.1016/j.neucom.2007.07.037.

Shobben, D., Torkkola, K. & Smaragdis, P. (1999), Evaluation of blind signal separation methods, *in Proc. of ICA and BSS*, pp. 239-244.

Smaragdis, P. (1998), Blind separation of convolved mixtures in the frequency domain, *in Proc. International workshop on Independence and Artificial Neural Networks.*

Solazzi, M., Piazza, F. & Uncini, A. (2000), An adaptive Spline Nonlinear Function for Blind Signal Processing, *in Proc. of IEEE Whorkshop on neural networks for signal Processing,* pp. 396-404.

Solazzi, M. & Uncini, A. (2000), Artificial neural networks with adaptive multidimensional spline activation functions, *in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000)*, vol. 3, pp. 471-476.

Solazzi, M., Piazza, F. & Uncini, A. (2001), Nonlinear Blind Source Separation by Spline neural Network, *in Proc.of ICASSP 2001*, pp. 2781-2784.

Solazzi, M. & Uncini, A. (2004), Regularizing Neural Networks using Flexible Multivariate Activation Function, *Neural Networks* **17**, 247-260.

Taleb, A. & Jutten, C. (1999), Source Separation in post nonlinear mixtures, *IEEE Trans. on Signal Processing* **47**(10), 2807-2820.

Taleb, A. (2002), A Generic Framework for Blind Source Separation in Structured Nonlinear Models, *IEEE Trans. on signal processing* **50**(8), 1819-1830.

Theis, F. J., Bauer, C. & Lang, E. (2002), Comparison of maximum entropy and minimal mutual information in a nonlinear setting, *Signal Processing* **82**, 971-980.

Theis, F.J. (2004a), A New Concept for Separability Problems in Blind Source Separation, *Neural Computation* **16**, 1827-1850.

Theis, F. J. (2004b), Uniqueness of complex and multidimensional independent component analysis, *Signal Processing* **84**, 951-956.

Theis, F. J. & Gruber, P. (2005), On model identifiability in analytic postnonlinear ICA, *Neurocomputing* **64**, 223-234.

Uncini, A., Vecci, L., Campolucci, P. & Piazza, F. (1999), Complex-valued Neural Networks with Adaptive Spline Activation Function for Digital Radio Links Nonlinear Equalization, *IEEE Trans. on Signal Processing* **47**(2), 505-514.

Uncini, A. & Piazza, F. (2003), Blind Signal Processing by Complex domain Adaptive Spline Neural Network, *IEEE Trans. on Neural Networks* **14**(2), 399-412.

Vecci, L., Piazza, F. & Uncini, A. (1998), Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks, *Neural Networks* **11**(2), 259-270.

Vigliano, D. & Uncini, A. (2003), Flexible ICA solution for a novel nonlinear blind source separation problem, *In IET Electronics Letters* **39**(22), 1616-1617.

Vigliano, D., Parisi, R. & Uncini, A. (2005), An Information Theoretic Approach to a Novel Nonlinear Independent Component Analysis Paradigm, *Elsevier Information Theoretic Signal Processing* **85**, 997-1028.

Vigliano, D., Scarpiniti, M., Parisi, R. & Uncini, A. (2006a), A flexible Blind source recovery in complex nonlinear environment, *in IEEE International Symposium on Intelligent Control*, pp. 3059-3063.

Vigliano, D., Scarpiniti, M., Parisi, R. & Uncini, A. (2006b), Flexible ICA in Complex and Nonlinear Environment by Mutual Information Minimization, *in Proc. IEEE Workshop on Machine Learning for Signal Processing*, pp. 59-63.

Vigliano, D., Scarpiniti, M., Parisi, R. & Uncini, A. (2008), Flexible Nonlinear Blind Signal Separation in the Complex Domain, *International Journal of Neural System* **18**(2), pp. 1-18.

Vitagliano, F., Parisi, R. & Uncini, A. (2003), Generalized Splitting 2D Flexible Activation Function, *in Lecture Notes in Computer Science*, Vol. 2859/2003, pp. 85-95.

Yang, H. H. & Amari, S. (1997), Adaptive Online Learning Algorithms for Blind Separation: Maximum Entropy and Minimum Mutual Information, *Neural Computation* **9**, 1457-1482.

ADDITIONAL READING

Adali, T. & Calhoun, V. (2007), Complex ICA of Brain Imaging Data, *IEEE Signal Processing Magazine*, **24**(5), 136-139.

Amari, S. & Cichocki, A. (1998), Adaptive Blind Signal Processing - Neural Network

Approaches, *in Proceedings of IEEE*, pp. 2026-2048.

Benesty, J. & Huang, (Ed.). (2003), *Adaptive Signal Processing*, Springer.

Benesty, J., Makino, S. & Chen, (Ed.). (2005), *Speech Enhancement*, Springer.

Cardoso, J. (1997), Infomax and maximum likelihood for blind source separation, *Signal Processing Letters, IEEE* **4**(4), 112-114.

Cardoso, J. & Comon, P. (1996), Independent component analysis, a survey of some algebraic methods, *in IEEE International Symposium on Circuits and Systems ISCAS '96*, Vol. 2, pp. 93-96.

Cardoso, J. (1998), Blind signal separation: statistical principles, *Proceedings of the IEEE* **86**(10), 2009-2025.

Cichocki, A. & Amari, S. (2002), *Adaptive Blind Signal and Image Processing,* John Wiley.

De Boor, C. (2001), *A practical guide to spline*, Springer.

Eriksson, J., Seppola, A. & Koivunen, V. (2005), Complex ICA for circular and non-circular sources, *in 13th European Signal Processing Conference EUSIPCO 2005*.

Girolami, M., (Ed.). (1999), *Self-organising Neural Networks: Independent Component Analysis and Blind Source Separation*, Springer.

Girolami, M. (2000), *Advances in Independent Component Analysis (Perspectives in Neural Computing),* Springer.

Hyvärinen, A., Karhunen, J. & Oja, E. (2001), *Independent Component Analysis,* John WIley & Sons, Inc..

Hyvarinen, A. (1999), Beyond independent components, *in Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, pp. 809-814vol.2.

Huang, Y., Benesty, J. & Chen, J. (2006), *Acoustic MIMO Signal Processing,* Springer.

Jutten, C. & Taleb, A. (2000), Source separation: from dusk till dawn, *in Proceeding of 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA'2000)*, pp. 15-26.

Lee, T.W. (1998), *Independent Component Analysis - Theory and Applications*, Springer.

Makino, S., Lee, T. & Sawada, H., (Ed.). (2007), *Blind Speech Separation*, Springer.

Mandic, D., Javidi, S., Souretis, G. & Goh, V. (2007), Why a Complex Valued Solution for a Real Domain Problem, *in Proc. IEEE Workshop on Machine Learning for Signal Processing,* pp. 384-389.

Novey, M. & Adali, T. (2007), Complex Fixed-Point ICA Algorithm for Separation of QAM Sources using Gaussian Mixture Model, *in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. II 445-448.

Picinbono, B. & Bondon, P. (1997), Second-order statistics of complex signals, *IEEE Transactions on Signal Processing,* **45**(2), 411-420.

Remmert, R. (1991), *Theory of Complex Functions,* Springer.

Schumaker, L.L. (2007), *Spline Functions: Basic Theory*, Cambridge Mathematical Library, 3-rd edition.

Stone, J. V. (2004), *Independent Component Analysis: A Tutorial Introduction,* Bradford Books.

Wahba, G. (1990), *Spline models for observational data*, Society for Industrial Mathematics.