



# An information theoretic approach to a novel nonlinear independent component analysis paradigm

Daniele Vigliano\*, Raffaele Parisi, Aurelio Uncini

*Dipartimento INFOCOM, Università di Roma "La sapienza", Via Eudossiana, 18, 00184 Rome, Italy*

Received 26 June 2004; received in revised form 29 September 2004

---

## Abstract

This paper introduces a novel independent component analysis (ICA) approach to the separation of nonlinear convolutive mixtures. The proposed model is an extension of the well-known post nonlinear (PNL) mixing model and consists of the convolutive mixing of PNL mixtures. Theoretical proof of existence and uniqueness of the solution under proper assumptions is provided. Feedforward and recurrent demixing architectures based on spline neurons are introduced and compared. Source separation is performed by minimizing the mutual information of the output signals with respect to the network parameters. More specifically, the proposed architectures perform on-line nonlinear compensation and score function estimation by proper use of flexible spline nonlinearities, yielding a significant performance improvement in terms of source pdf matching and algorithm speed of convergence. Experimental tests on different signals are described to demonstrate the effectiveness of the proposed approach.

© 2005 Elsevier B.V. All rights reserved.

*Keywords:* Mutual information; Kullback–Leibler divergence; Blind source separation; Independent component analysis; Nonlinear ICA; Feedforward and recurrent networks; Flexible activation functions; Spline functions; On-line learning

---

## 1. Introduction

The interest of the scientific community for blind signal processing and in particular for blind source separation (BSS), performed through independent component analysis (ICA), has been considerably growing in last years. This interest is justified by the number of different approaches and applications. As a matter of fact, in several fields, from multimedia to telecommunications and biomedicine, ICA is currently employed to effectively remove interfering signals from the signal of interest. Furthermore it is interesting

---

\*Corresponding author. Via Carlo de Marchesetti 84, 00172 Rome, Italy. Tel.: +39 06 85099551.

*E-mail addresses:* Daniele.vigliano@poste.it (D. Vigliano), parisi@infocom.uniroma1.it (R. Parisi), aurel@infocom.uniroma1.it (A. Uncini).

to note that many array signal processing or blind channel deconvolution problems can now be casted into the ICA framework.

Initial studies on ICA aimed at solving the well-known *cocktail party problem*, in a static or slightly reverberant environment. Pioneering works in ICA appeared at the beginning of the 1990s, when Jutten and Herault presented their “neurometric architecture” [1] and Comon published his often referenced paper [2]. From that time many different algorithms have been developed. The main approaches to ICA were by nonlinear principal component analysis (PCA), factor analysis, projection pursuit and redundancy reduction. Some approaches to source separation were based on estimation of high order cumulants [2,3]. The INFOMAX principle inspired the well-known MaxEntropy method [4], while other solutions to ICA employed the minimization of mutual information (MMI) [5] or the maximization of the likelihood function [6,7]. A very popular algorithm is FastICA, which minimizes the negentropy using a fixed point algorithm [8] and finally the EASI algorithm [9] uses the so-called “relative gradient” instead of a stochastic gradient in the MMI framework. All these algorithms are based on different cost functions, but can be embodied in a common general formulation [10].

One of the more critical issues in ICA is the matching between the probability distribution function (pdf) of sources and the algorithm’s parameters. In order to improve the pdf matching process, the so-called Flexible ICA was recently proposed [11]. Flexible ICA provides faster learning by estimating the parameters related to the pdf of signals. Several methods based on polynomials [12] and on parametric function approaches [7,13] were proposed.

Conventional ICA approaches perform the blind source recovery in static or slightly reverberant environment. Unfortunately linear mixing models are too unrealistic and unsatisfactory in a lot of real situations. Examples of more complex nonlinear mixing models were presented in [14,15]. In [16] the existence of the solution to some nonlinear problems was explored. Nonuniqueness of the solution in the general case was evidenced, unless some additive constraints about the mixing model are taken into account.

Recently the so-called post nonlinear (PNL) mixing problem has received much attention, also with regard to the existence and uniqueness problem [17–20]. Only a few works have addressed the convolutive PNL [21,22] or even a more complex static nonlinear mixing [23]. In [24] a complete review of most recent results in BSS of nonlinear mixing models was presented.

In this paper, the solution of the BSS problem in a novel convolutive nonlinear mixing environment is introduced and described. Some results were preliminary described in [25]. The proposed model has a higher complexity of the ones currently available in literature, being composed by a PNL block followed by a convolutive mixing channel. In particular the theoretical proof of the existence and the uniqueness of the solution under some assumptions is furnished. Different separating architectures based on the use of feedforward and recurrent networks are tested and compared in typical mixing environments.

## 2. Nonlinear convolutive mixing model

The aim of this section is to describe the existing approaches to the BSS problem in different mixing environments and to briefly explore the main results in terms of existence and uniqueness of the solution.

Given a set of  $N$  source signals at time  $n$   $\mathbf{s}[n] = \{s_1[n], \dots, s_N[n]\}^T$ , conventional ICA tries to recover  $\mathbf{s}[n]$  from the observation of a linear at least convolutive mixture  $\mathbf{x}[n] = \{x_1[n], \dots, x_N[n]\}^T$ . The solution can be found up to some trivial nonuniqueness and desired solution can be expressed as

$$\mathbf{y}[n] = \mathbf{P}\mathbf{A}\mathbf{D}\mathbf{s}[n], \quad (1)$$

where  $\mathbf{P}$  is a permutation matrix,  $\mathbf{A}$  is a diagonal scaling matrix and  $\mathbf{D}$  is a diagonal delay matrix.

Conventional ICA must be properly extended in order to take into account the higher complexity of real environments. For this reason in last years more complex mixing environments have been introduced. The general formulation of a nonlinear hidden mixing model in a convolutive environment is

$$\mathbf{x}[n] = \mathcal{F}\{\mathbf{s}[n], \dots, \mathbf{s}[n - L]\}, \tag{2}$$

where  $\mathcal{F}\{\cdot\}$  is some nonlinear functional. Given the mixing environment (2), the solution of the BSS problem is found by a proper transformation  $\mathcal{G}\{\cdot\}$

$$\mathbf{y}[n] = \{y_1[n], \dots, y_N[n]\}^T = \mathcal{G}\{\mathbf{x}[n]\}. \tag{3}$$

The issue of recovering the original sources in the general mixing model (2) with no particular a priori assumptions is affected by a strong nonuniqueness. To illustrate this, consider two independent random variables  $s_1$  with uniform distribution in  $[0, 2\pi]$  and  $s_2$  with Rayleigh distribution. Given the two nonlinear transformations  $y_1 = s_2 \cos s_1$  and  $y_2 = s_2 \sin s_1$ , the random variables  $y_1, y_2$  are still independent but are gaussian distributed [23]. 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  $\mathcal{F}\{\cdot\}$  or the mixing and demixing model are made [18]. In practice the main issue is to find the theoretical conditions in terms of sources, mixing environment, recovering architecture, capable to guarantee the existence of the solution. For example in [16] a constructive approach to the separation problem in a general static nonlinear mixing environment was proposed. In this work a Gram–Schmidt procedure is employed to iteratively separate each component of the mixed signals. Uniqueness of the solution (up to some trivial nonuniqueness) is guaranteed if the number of sources is two, the mixing function  $\mathcal{F}\{\cdot\}$  is a conformal zero preserving mapping and the densities of the independent components have a bounded known support.

A possible conformal mapping is the well-known PNL model [17–19,23]. In this case a proof of the existence and uniqueness of the solution with more than two sources was furnished [18]. In addition, a PNL convolutive mixing model was studied [22], also when a static mixing after the PNL model is added [23].

The solution of the general nonlinear convolutive ICA requires proper a priori assumptions concerning the mixing–demixing model. Using the same notation of (2) and (3), let  $\mathcal{Y}$  be the set of all vectors  $\mathbf{y}$  having independent components, defined by

$$Y = \left\{ \mathbf{y} \mid p_{\mathbf{y}}(\mathbf{y}) = \prod_i p_{y_i}(y_i); \mathbf{y} = \mathcal{G} \circ \mathbf{x} = \mathcal{G} \circ \mathcal{F} \circ \mathbf{s} = \mathcal{H} \circ \mathbf{s} \right\}, \tag{4}$$

where  $\mathcal{H}\{\cdot\}$  in general has a nondiagonal Jacobian matrix.

As a matter of fact, it is possible to find an infinite number of functions  $\mathcal{G}\{\cdot\}$  such that  $\mathbf{y} = \mathcal{G} \circ \mathbf{x} \in \mathcal{Y}$ , but not all of them have a diagonal Jacobian matrix. So most of the solutions in  $\mathcal{Y}$  are not of interest, to mean that output independence by itself is a weak approach to the BSS problem in a general nonlinear environment.

In the following the issue of recovering original sources in the presence of a nonlinear convolutive mixing is explored. A priori knowledge of the mixing model is exploited to design the recovering network.

### 2.1. Model description

In this work the convolutive nonlinear mixing model of Fig. 1 is assumed.

In Fig. 1  $\mathbf{A}$  is a  $N \times N$  matrix,  $\mathbf{F}[\mathbf{r}[n]] = [f_1[r_1[n]], \dots, f_N[r_N[n]]]^T$  is a  $N \times 1$  vector of nonlinear functions (one for each channel) and  $\mathbf{Z}[k]$  is a FIR matrix where each element is a  $L_Z$ -tap FIR filter [26].

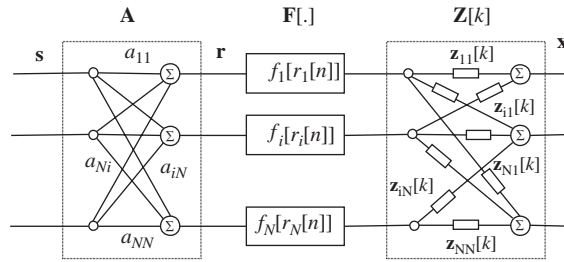


Fig. 1. Block diagram of the proposed convolutive nonlinear mixing model.

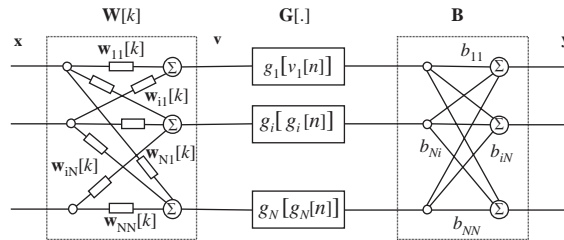


Fig. 2. Block diagram of the recovering architecture.

In formulas

$$\mathbf{x}[n] = \mathcal{F}[s[n]] = \sum_{k=0}^{L_Z-1} \mathbf{Z}[k]\mathbf{F}[\mathbf{A}s[n-k]]. \tag{5}$$

Interest in this novel mixing environment is justified mainly by its higher generality. Moreover the PNL model can be considered as a particular case of (5).

The recovering architecture is represented in Fig. 2 and is based on the so-called *mirror model* [27].

In Fig. 2 matrix  $\mathbf{W}$ , vector  $\mathbf{G}$ , and matrix  $\mathbf{B}$  perform channel inversion, nonlinear compensation and static demixing, respectively. If  $\mathbf{W}$  is implemented by a FIR matrix where each element is a  $L_W$ -tap FIR filter, the outputs of the recovering structure can be written as

$$\mathbf{y}[n] = \mathcal{G}[\mathbf{x}] = \mathbf{B}\mathbf{G}\left[\sum_{k=0}^{L_W-1} \mathbf{W}[k]\mathbf{x}[n-k]\right]. \tag{6}$$

Knowledge about the mixing model is the key to avoid the strict nonuniqueness of the solution, in the sense of limiting the cardinality of all possible independent output solutions.

### 2.2. Source separability

The following proposition extends Lemma 1 formerly introduced in [19] and shows how the elements of  $\mathcal{Y}$  can differ only for some trivial ambiguity, if the mixing model is (5) and the demixing model is (6).

**Proposition 1.** *Given the convolutive, nonlinear mixture model  $\mathcal{F}\{\mathbf{A}, \mathbf{F}, \mathbf{Z}\}$  (5) and the recovery model  $\mathcal{G}\{\mathbf{B}, \mathbf{G}, \mathbf{W}\}$  (6), assuming that:*

- (a)  $\mathbf{A}$  is a nonsingular matrix of nonzero entries;
- (b)  $\mathbf{Z}[k]$  is a convolutive mixing channel admitting an inverse;

- (c)  $g_i$  and  $f_i$  ( $i = 1, \dots, N$ ) are differentiable, invertible, zero preserving monotone functions;
- (d)  $\mathbf{s}[n]$  is a random vector whose components are spatially independent, temporally white and have finite support;
- (e) the pdf of  $s_i[n]$  vanishes for at least one real  $\bar{s}_i$ ;

then the components of the output vector  $\mathbf{y}[n]$  are independent if and only if

$$\mathbf{y}[n] = \mathbf{P}\mathbf{A}\mathbf{D}\mathbf{s}[n] = \mathbf{P} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix} \begin{bmatrix} z^{-d} & & 0 \\ & \ddots & \\ 0 & & z^{-d} \end{bmatrix} \mathbf{s}[n]. \quad (7)$$

**Proof.** See the Appendix A.

In Proposition 1 the convolutive environment forces to model inputs as stochastic processes so signals  $s_i[n]$  and  $s_j[m]$  are spatially independent if and only if  $p_{s_i s_j}(s_i[n], s_j[m]) = p_{s_j}(s_j[m])p_{s_i}(s_i[n]) \forall n, m$ .

Assuming that the mixing environment is invertible, Proposition 1 guarantees the existence and the uniqueness of the solution of the BSS problem in the presence of white sources and FIR demixing matrices. In addition, experimental tests showed that good results were obtained also when an IIR demixing model was used to separate speech signals.

### 3. Demixing algorithms and architectures

The efficient design of the demixing strategy requires the choice of the proper demixing model, the cost function measuring independence of the outputs and an effective optimization method. In this section feedforward and recurrent networks will be proposed and investigated as effective demixing models. Network parameters will be iteratively adapted (i.e. learned) on the basis of a measure of the output independence. In particular, the independence of the outputs will be measured by considering the Kullback–Leibler (KL) divergence between  $p_{\mathbf{Y}}[\mathbf{y}]$  and  $\tilde{p}_{\mathbf{Y}}[\mathbf{y}] = \prod_{i=1}^N p_{y_i}[y_i]$ . Considering a demixing model with parameters  $\Phi$ , the cost function to be minimized is

$$\mathcal{B}\{\mathbf{y}[n], \Phi\} = \text{KL} \left( p_{\mathbf{Y}}, \prod_{i=1}^N p_{y_i}[y_i] \right) = \int_{\mathfrak{Y}} p_{\mathbf{Y}}(\mathbf{y}) \log \left( p_{\mathbf{Y}}(\mathbf{y}) / \prod_{i=1}^N p_{y_i}[y_i] \right) d\mathbf{y} = \sum_{i=1}^N H(y_i) - H(\mathbf{y}) = I(\mathbf{y}). \quad (8)$$

The proposed expression of KL divergence (8) is equivalent to the mutual information  $I(\mathbf{y})$  among the components of  $\mathbf{y}[n]$  and is a measure of independence well known in BSS field [2,5,10,17–19]. Other approaches, instead of minimizing the output mutual information, maximize the output entropy  $H(\mathbf{y})$ . Anyway these methods have a serious limitation, since entropy maximization does not necessarily lead to mutual information minimization. In particular, the KL divergence is preferred for the following attractive properties:

- (a)  $\text{KL}(p_{\mathbf{Y}}, \tilde{p}_{\mathbf{Y}}) \geq 0$ ;  $\text{KL}(p_{\mathbf{Y}}, \tilde{p}_{\mathbf{Y}}) = 0 \Leftrightarrow p_{\mathbf{Y}}(\mathbf{y}) \equiv \prod_{i=1}^N p_{y_i}(y_i)$ ;
- (b)  $\text{KL}(p_{\mathbf{Y}}, \tilde{p}_{\mathbf{Y}})$  is invariant with respect to permutation, scaling and transformation of the outputs by a nonlinear monotone function;
- (c) in this formulation KL divergence is equivalent to the mutual information  $I(\mathbf{y})$  that has a symmetric structure [28]:  $I(y_1, y_2) = I(y_2, y_1)$ .

The *stochastic gradient* and the *natural gradient* [29] were used to minimize the KL divergence with respect to the model parameters. In particular the advantages of using the natural gradient have been extensively

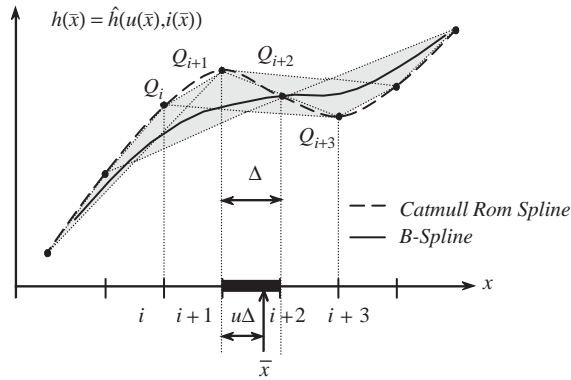


Fig. 3. Example of a spline interpolation of four control points.

described in literature [5,29–36]. The choice of a gradient-based procedure leads to consider the derivatives of (8) with respect to the model parameters

$$\frac{\partial}{\partial \Phi} \log[p_{y_i}(y_i)] = \frac{\dot{p}_{y_i}(y_i)}{p_{y_i}(y_i)} \frac{\partial y_i}{\partial \Phi} = \psi_i(y_i) \frac{\partial y_i}{\partial \Phi}, \tag{9}$$

where  $\psi_i(y_i) = \dot{p}_i(y_i)/p_i(y_i)$  is the so-called *score function* (SF). Estimation of  $\psi_i(y_i)$  is a critical step for the network learning, as described in detail in Section 3.1.1.

In the following two neural feedforward and recurrent architectures for BSS are proposed. They are both based on the use of the *spline neuron* as a basic *flexible* building block.

### 3.1. Estimation of nonlinear functions: the spline neuron

#### 3.1.1. Spline approximation neuron

Splines are smooth parametric curves defined by interpolation of properly defined control points collected in a lookup table. This section briefly summarizes the main properties of splines and their use into the neural net framework [22,37–41].

Let  $y = h(x)$  be some function to be estimated. The spline estimation neuron provides an approximation  $h(x) \cong \hat{y} = \hat{h}(u(x), i(x))$  based on two parameters ( $u, i$ ) directly depending on  $x$ . In the general case, given  $N$  equispaced control points, the spline curve results as a polynomial interpolation through  $N - 1$  adjacent spans. In this specific application, for each input occurrence  $x$  the spline neuron estimates  $h(x)$  by using four control points selected inside the lookup table. Two points are the adjacent control points on the left side of  $x$ , while the other two points are the two control points on the right side; Fig. 3 shows an example of interpolation.

If  $i$  is the index of the leftmost of the four control points and  $u$  ( $0 < u \leq 1$ ) is the local<sup>1</sup> abscissa, the pair  $(i, u)$  is computed using a dummy variable  $z$  as follows

$$\begin{cases} i = \lfloor z \rfloor \\ u = z - i \end{cases} z : z = \begin{cases} 1 & \alpha \leq 1 \\ z_l & 1 < \alpha \leq N - 3 \\ N - 3 & \alpha > N - 3 \end{cases} \text{ where } \alpha = \frac{x}{\Delta} + \frac{N - 1}{2}, \tag{10}$$

<sup>1</sup>Variable  $u$  is local with respect to the distance between two adjacent control points.

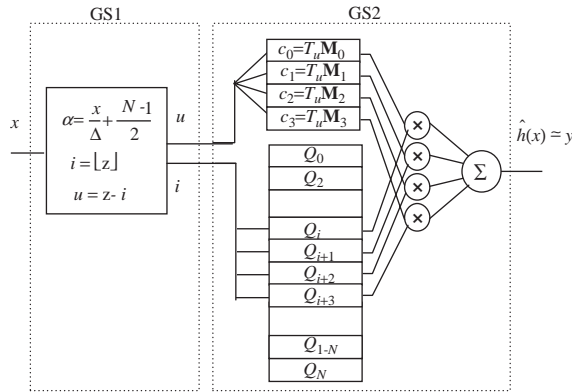


Fig. 4. Example of a spline interpolation of four control points.

where  $\Delta$  is the length of each span. The complete expression of the spline estimation model is

$$y = \hat{h}(u, i) = 1/2\mathbf{T}_u\mathbf{M}\mathbf{Q}_i, \tag{11}$$

where

$$\mathbf{T}_u = [u^3 \ u^2 \ u^1 \ 1], \tag{12}$$

$\mathbf{Q}_i = [Q_i \ Q_{i+1} \ Q_{i+2} \ Q_{i+3}]$  contains the  $y$ -values of the four control points and  $\mathbf{M} \in \mathcal{M}_{4,4}$  (i.e. the space of the square  $4 \times 4$  matrices). Different kinds of spline functions (e.g. Catmull-Rom or B-Spline) are available, that differ for the choice of matrix  $\mathbf{M}$ . In particular, an attractive property of the Catmull-Spline is that control points lie exactly on the estimated function. In this case  $\mathbf{M}$  is

$$\mathbf{M} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}. \tag{13}$$

Fig. 4 represents the structure of the spline adaptive network, modelled with two blocks: GS1 and GS2.

The learning algorithm of the spline neuron is local and independent of the number of control points, since only four control points are involved for each learning sample. This is the main advantage of using the spline neural net with respect to other architectures like polynomial networks or multilayer perceptrons (MLP).

The number of control points is not a critical issue in the approximation capabilities but must be properly chosen. In fact it is well-known that a high number of control points can produce overfitting and underlearning, while a reduced number of control points can cause underfitting and overlearning.

### 3.1.2. Spline approximation for ICA: direct estimation of score functions

SFs<sup>2</sup> (9) can be used to define the following vector:

$$\Psi_{\mathbf{y}}[\mathbf{y}] = [\psi_{y_1}(y_1) \cdots \psi_{y_N}(y_N)]^T = [\dot{p}_{y_1}(y_1)/p_{y_1}(y_1) \cdots \dot{p}_{y_N}(y_N)/p_{y_N}(y_N)]^T. \tag{14}$$

<sup>2</sup>Sometimes score functions are defined in literature as in (14) with minus sign.

In the totally blind case there is no a priori information on the hidden sources or output pdf. This is the reason why the output pdf should be estimated only during the learning phase, since output signals may change.<sup>3</sup> Using a predetermined SF based on some a priori estimation is theoretically possible but leads to worse convergence performance. As a matter of fact, the matching between signals' pdf and the corresponding SFs is a critical issue for the learning algorithm, since it determines the performance in separation. In [18] the Gram–Charlier approximation was compared to the MLP estimator in estimating the pdf and the SFs. In [12] a polynomial function with adaptively learning coefficients was proposed. In [7,13] a linear parametric estimation model based on a projection in a subspace spanned by nonlinear functions was described. All these approaches are limited by the fact that learning is not local and in several cases is performed off-line.

In [13] direct estimation of SFs by the least mean square (LMS) algorithm was described. Parameters were estimated by minimizing the mean square error  $\varepsilon_j$  for each output channel  $j$

$$\varepsilon_j = \frac{1}{2} E\{[\tilde{\psi}_j(y_j, \mathbf{\Phi}) - \dot{p}_{y_j}(y_j)/p_{y_j}(y_j)]^2\}, \quad j = 1 \dots N. \quad (15)$$

In (15)  $\tilde{\psi}_j(y_j, \mathbf{\Phi})$  is the spline model of the SF, while  $E\{\cdot\}$  is the expectation operator.

The gradient of (15) with respect to parameters  $\mathbf{Q}^\psi$  of the spline model is

$$\frac{\partial \varepsilon_j}{\partial \mathbf{Q}^{\psi_i}} = E \left\{ \frac{\partial \tilde{\psi}_j}{\partial \mathbf{Q}^{\psi_i}} \left( \tilde{\psi}_j - \frac{\dot{p}_{y_j}(y_j)}{p_{y_j}(y_j)} \right) \right\}. \quad (16)$$

Assuming that  $y_j$  is a random variable and  $\psi_j(y_j)$  is the theoretical SF, if  $f$  is a differentiable function satisfying  $\lim_{|y_j| \rightarrow \infty} p_{y_j}(y_j)f(y_j) = 0$  then

$$E[f(y_j)\psi_j(y_j)] = -E[\dot{f}(y_j)] \quad (17)$$

(see [19] for the proof). Applying (17) to (15) we obtain

$$\frac{\partial \varepsilon_j}{\partial \mathbf{Q}^{\psi_i}} = \frac{\partial \tilde{\psi}_j(y_j, \mathbf{Q}^{\psi_i})}{\partial \mathbf{Q}^{\psi_i}} \tilde{\psi}_j + \frac{\partial^2 \tilde{\psi}_j(y_j, \mathbf{Q}^{\psi_i})}{\partial y_j \partial \mathbf{Q}^{\psi_i}}. \quad (18)$$

Eq. (18) does not contain the unknown pdf anymore but only the estimation model [19], thus making it possible to use an unsupervised optimization algorithm. Optimization can be performed by the conventional steepest descent method:

$$\mathbf{Q}^\psi(k+1) = \mathbf{Q}^\psi(k) - \eta_{\mathbf{Q}^\psi}(k) \frac{\partial}{\partial \mathbf{Q}^\psi} \varepsilon\{\mathbf{\Phi}(k), \mathbf{y}\}. \quad (19)$$

Derivation of the cost function yields the gradient expression for the spline control points:

$$\frac{\partial \varepsilon}{\partial \mathbf{Q}^\psi} = \left[ \frac{1}{4} \mathbf{T}_U \mathbf{M} \mathbf{T}_U \mathbf{M} \mathbf{Q}^\psi + \frac{1}{\Delta} \dot{\mathbf{T}}_U \mathbf{M} \right], \quad (20)$$

where  $\Delta$  is the difference between the abscissas of adjacent control points.

### 3.2. Feedforward spline networks

Having explored the mixing model and the associated cost function, the next step is to define the recovering network and to derive its learning rules. The proposed feedforward architecture is made of a cascade of blocks and is depicted in Fig. 5 in the case of two sources.

<sup>3</sup>Note that during the learning phase output signals can also flip.



Fig. 5 shows the two blocks (SG1 and SG2) spline neurons involved in the compensation of nonlinear inversion (see Fig. 4 for the architecture of each block).

Several examples of FIR-based architectures are available in literature [4,5,8,9,18,21,26,42–45]. The critical issue of these structures is the filter length, since a high number of taps is required to invert real channels, with the consequence that the computational cost grows exponentially. In the proposed architecture, the output of the generic  $i$ th channel at time  $n$  is

$$y_i[n] = \sum_{j=1}^N b_{ij}g_j \left[ \sum_{h=1}^N \sum_{k=0}^{L-1} w_{jh[k]}x_h[n-k] \right].$$

Let define the input vector at time  $n$   $\mathbf{x}[n] = [x_1[n], \dots, x_N[n]]$  and the corresponding output vector  $\mathbf{y}[n]$ . For  $n = 0, \dots, M$  we define the vectors  $\mathcal{X} = [\mathbf{x}[0], \dots, \mathbf{x}[M]]^T$  and  $\mathcal{Y} = [\mathbf{y}[0], \dots, \mathbf{y}[M]]^T$  and the output pdf as follows:

$$p_y(y) = \frac{p_x(x)}{J} = \frac{p_x(x)}{\prod_{n=0}^M \left\{ |\det \mathbf{B}| \left( \prod_{i=1}^N |g_i[v_i[n]]| \right) |\det \mathbf{W}[0]| \right\}}, \quad (21)$$

where  $J$  is the determinant of the Jacobian matrix of the transformation between  $\mathcal{X}$  and  $\mathcal{Y}$  (see Appendix B for the derivation of (21)). Indicating by  $\Phi_G$  and  $\Phi_\Psi$  the parameters of the splines used to estimate the SFs and the compensating nonlinear functions, the complete set of learning parameters can be expressed by

$$\Phi = \{b_{ij}, w_{pq}[k], \Phi_G, \Phi_\Psi \quad | \forall i, j, p, q; k = 0 \dots L - 1\}.$$

Eq. (8) can be rewritten by replacing  $p_Y[\mathbf{y}]$  with (21), getting a new functional  $\mathfrak{J}(\Phi, y)$ . Finally, derivation with respect to  $\Phi$  leads to

$$\begin{aligned} \frac{\partial}{\partial \Phi} \mathfrak{J}(\Phi, y) &= \frac{\partial}{\partial \Phi} \sum_{n=0}^M \left[ -\log |\det \mathbf{B}| - \log \prod_{i=1}^N g_i[v_i[n]] - \log |\det \mathbf{W}[0]| - \sum_{i=1}^N \log p_{y_i}(y_i) \right]. \end{aligned} \quad (22)$$

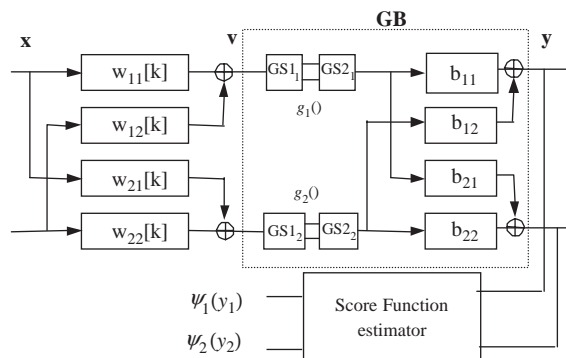


Fig. 5. Proposed feedforward network for nonlinear blind deconvolution and separation.

In (22) expected values have been replaced by instantaneous values. Application of the steepest descent method yields

$$\Phi(l+1) = \Phi(l) - \eta_{\Phi}(l) \frac{\partial}{\partial \Phi} \mathfrak{I}\{\Phi(l), y\}, \quad (23)$$

where  $\Phi(l)$  is the parameter set at the  $l$ th iteration. In particular, the gradient with respect to the elements of  $\mathbf{B}$  is

$$\frac{\partial}{\partial \mathbf{B}} \mathfrak{I}\{\Phi, y\} = -\mathbf{B}^{-T} - \Psi_y \mathbf{G}[\mathbf{v}]^T. \quad (24)$$

The natural gradient can be obtained by right multiplication by  $\mathbf{B}^T \mathbf{B}$ .

The gradient with respect to the elements of the FIR matrix  $\mathbf{W}$  is

$$\begin{aligned} & \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{W}[k]} \\ &= -\mathbf{W}[0]^{-T} \delta_k - \left\{ \begin{bmatrix} \ddot{g}_1(v_1)/\dot{g}_1(v_1) \\ \vdots \\ \ddot{g}_N(v_N)/\dot{g}_N(v_N) \end{bmatrix} + \begin{bmatrix} \Psi_y^T(\mathbf{B})_1 & 0 \\ 0 & \Psi_y^T(\mathbf{B})_N \end{bmatrix} \begin{bmatrix} \dot{g}_1(v_1) \\ \vdots \\ \dot{g}_N(v_N) \end{bmatrix} \right\} \mathbf{x}^T(n-k), \end{aligned} \quad (25)$$

where  $(\mathbf{B})_m$  is the  $m$ th row of the matrix  $\mathbf{B}$ .

The natural gradient can be obtained, considering the advice of Amari [36], by right applying the term  $\mathbf{W}^T[k] \mathbf{W}[k]$  to (25) as follows:

$$\begin{aligned} & \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{W}[k]} \\ &= -\mathbf{W}[k] \delta_k - \begin{bmatrix} \ddot{g}_1(v_1)/\dot{g}_1(v_1) + \Psi_y^T(\mathbf{B})_1 \dot{g}_1(v_1) \\ \vdots \\ \ddot{g}_N(v_N)/\dot{g}_N(v_N) + \Psi_y^T(\mathbf{B})_N \dot{g}_N(v_N) \end{bmatrix} \sum_{q=0}^{L_W-1} \mathbf{r}^T(n-k+q) \mathbf{W}[q]. \end{aligned} \quad (26)$$

Eq. (26) is noncausal for lower values of  $k$ . Causality can be recovered by introducing a proper delay  $d$  on the gradient computation. The steepest descent causalized learning rule becomes

$$\mathbf{W}(l+1) = \mathbf{W}(l) - \eta_{\mathbf{W}} \frac{\partial}{\partial \mathbf{W}} \mathfrak{I}\{\Phi(l-d), y\}, \quad (27)$$

where  $d$  is set on the basis of the FIR size [30,36] and  $\mathbf{W}$  has been used for  $\mathbf{W}[k]$ . Some solutions exist where both sides of (25) are right multiplied by  $\mathbf{W}^T[0] \mathbf{W}[0] \delta_k$ , thus avoiding matrix inversion [30].

The gradient with respect to the control points  $\mathbf{Q}^g$  of the spline compensating for the nonlinear distorting functions is

$$\frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{Q}_i^{g_j}} = - \left[ \frac{\dot{\mathbf{T}}_u \mathbf{M}}{\dot{\mathbf{T}}_u \mathbf{M} \mathbf{Q}_i^{g_j}} + \Psi_y^T(\mathbf{B})_j \mathbf{T}_u \mathbf{M} \right]. \quad (28)$$

The learning rule for the spline neurons dedicated to the SFs can be derived from (20). The most attractive property of the natural gradient is the equivariance for which the learning is independent with respect to the initial conditions. In addition, the natural gradient provides more stable learning and faster convergence when compared to the stochastic gradient [29,31,36].

As already noted, the effectiveness of using FIR matrices is hampered by the length of filters required in practical problems. In fact, the time required for learning grows exponentially with the filter length. Preliminary results on this issue were provided in [25].

### 3.3. Recurrent spline networks

Recurrent spline networks offer a more compact alternative to FIR-based architectures. MIMO recurrent nets have been already applied in the past to the problem of signal deconvolution [43,46–48,32–35].

In this work the hybrid architecture presented in [35] has been adapted to the novel mixing model. Fig. 6 shows the proposed network when two inputs are considered. The block named **GB** has the same structure as the output **GB** block of Fig. 5.

The output of the network is

$$y_i[n] = \sum_{j=1}^N b_{ij}g_j[v_j[n]], \tag{29}$$

where

$$v_j[n] = \sum_{h=1}^N p_{jh}x_h[n] + \sum_{\substack{h=1 \\ h \neq j}}^N \sum_{k=1}^L w_{jh}[k]v_h[n-k].$$

Using the same notation of previous section, the output pdf can be expressed as

$$p_y(y) = \frac{p_x(x)}{J} = \frac{p_x(x)}{\prod_{n=0}^M \left\{ |\det \mathbf{B}| |\det \mathbf{P}| \left( \prod_{i=1}^N |g_i[v_i[n]]| \right) \right\}}. \tag{30}$$

The set of learning parameters is  $\Phi = p_{nm}, b_{ij}, w_{rt}[k], \Phi_G, \Phi_\psi \forall i, j, n, m, r, t; k = 0 \dots L - 1$ , where  $\Phi_G$  and  $\Phi_\psi$  have the same meaning as before.

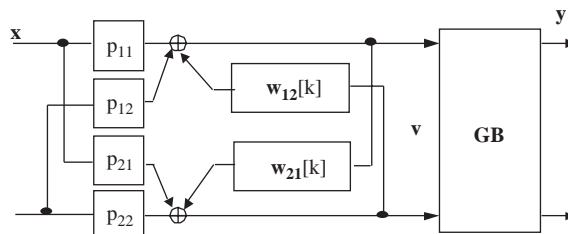


Fig. 6. Recurrent network for nonlinear blind deconvolution and separation.

Substitution of (30) into (8) and derivation with respect to parameters gives

$$\begin{aligned} & \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \Phi} \\ &= \frac{\partial}{\partial \Phi} \sum_{n=0}^M \left[ -\log |\det \mathbf{B}| - \log |\det \mathbf{P}| - \log \prod_{i=1}^N g_i[v_i[n]] - \sum_{i=1}^N \log p_{y_i}(y_i) \right] \end{aligned} \quad (31)$$

which is a little different from (22).<sup>4</sup>

The learning rules for the elements of  $\mathbf{B}$  are the same of (24) for both the stochastic and the natural gradients, while the learning rule for the spline neural network that estimates the nonlinear function is derived from (28). In addition, the gradient with respect to the elements of  $\mathbf{P}$  is

$$\begin{aligned} & \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{P}} \\ &= -\mathbf{P}^{-T} - \left\{ \begin{bmatrix} \ddot{g}_1(v_1)/\dot{g}_1(v_1) \\ \vdots \\ \ddot{g}_N(v_N)/\dot{g}_N(v_N) \end{bmatrix} + \begin{bmatrix} \Psi_y^T(\mathbf{B})_1 & 0 \\ 0 & \Psi_y^T(\mathbf{B})_N \end{bmatrix} \begin{bmatrix} \dot{g}_1(v_1) \\ \vdots \\ \dot{g}_N(v_N) \end{bmatrix} \right\} \mathbf{x}^T[n]. \end{aligned} \quad (32)$$

Right multiplication of (32) by  $\mathbf{P}^T \mathbf{P}$  gives the natural gradient. The gradient with respect to the IIR filter taps  $\mathbf{w}$  is

$$\begin{cases} \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{w}_{12}[k]} = -\left( \frac{\ddot{g}_1(v_1)}{\dot{g}_1(v_1)} + \Psi_y^T(\mathbf{B})_1 \dot{g}_1(v_1) \right) v_2[n-k-1], \\ \frac{\partial \mathfrak{I}\{\Phi, y\}}{\partial \mathbf{w}_{21}[k]} = -\left( \frac{\ddot{g}_2(v_2)}{\dot{g}_2(v_2)} + \Psi_y^T(\mathbf{B})_2 \dot{g}_2(v_2) \right) v_1[n-k-1], \end{cases} \quad k = 0 \dots L-1, \quad (33)$$

where  $(\mathbf{B})_i$  is the  $i$ th column of matrix  $\mathbf{B}$ . The gradient for the spline model of the SF is again (20).

#### 4. Experimental results

Several experimental tests were performed to assess the performance of the proposed architectures. Different solutions with different mixing environments were considered and compared. In order to make it possible the proper visualization of results, only pairs of mixed signals were considered.

Signals were assumed to lie in the range  $[-1, 1]$  and normalized so that signals  $v_1$  and  $v_2$  in Fig. 2 span the range  $[-0.8, 0.8]$ . Different indexes of performance to evaluate the output separation are available in literature [1,6,7]. In this paper the separation index  $S_j$  of the  $j$ th source was adopted [49]

$$S_j = 10 \log \left[ \frac{E\{(y_{\sigma(j),j})^2\}}{E\left\{ \sum_{k \neq j} (y_{\sigma(j),k})^2 \right\}} \right]. \quad (34)$$

In (34)  $y_{i,j}$  is the  $i$ th output signal when only the  $j$ th input signal is present while  $\sigma(j)$  is the output channel corresponding to the  $j$ -input.

<sup>4</sup>All considerations previously done about (22) are still applicable.

**Experiment 1.** The first test was used to verify the validity of Proposition 1. Two white random signals having Gaussian and uniform distributions, respectively, were considered.

The mixing matrix was  $\mathbf{A} = [0.6, -0.3; 0.5, 0.7]$ , the nonlinear distortion  $\mathbf{F}[f_1(r_1), f_2(r_2)] = [r_1 + 2r_1^3, 0.5r_2 + \tanh(7r_2)]$ , while the convolutive channel was represented by

$$\mathbf{Z}[k] = \begin{bmatrix} 0.4 - 0.15z^{-1} + 0.3z^{-2} & 0.15 + 0.1z^{-1} - 0.03z^{-2} \\ -0.15 + 0.15z^{-1} + 0.05z^{-2} & 0.35 + 0.15z^{-1} - 0.05z^{-2} \end{bmatrix}.$$

According to Proposition 1, separation was performed by the feedforward network. Fig. 7 shows the joint distributions of the source, the mixed and the recovered signals.

Fig. 8 shows the separation indexes for the two sources during learning. Separation is reached after about 150 epochs.

**Experiment 2.** More tests were performed in extension to conditions of Proposition 1. In particular the use of flexible on-line SF estimation allows to deal with both subgaussian and supergaussian sources, with no assumptions on the statistical properties of the sources. The following results are thus representative of a large number of real situations.

Three signals were considered in the following tests: a male speech  $s_1$ , a female speech  $s_2$  and white gaussian noise  $s_3$ . These signals are shown in Fig. 9, together with their empirical distributions. Fig. 10 shows the joint pdf of pairs  $[s_1, s_2]$  and  $[s_2, s_3]$ .

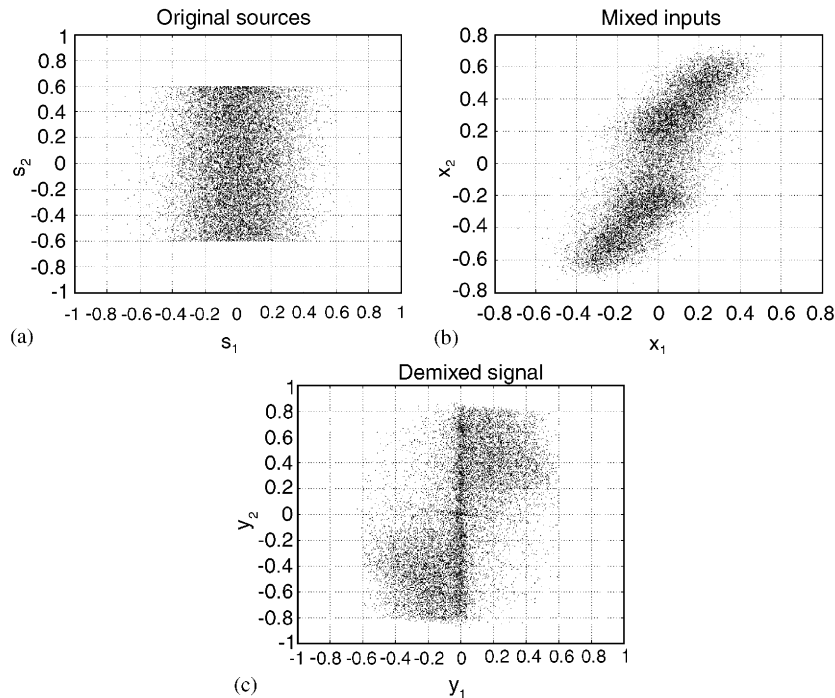


Fig. 7. Experiment 1: Gaussian ( $s_1$ ) and uniformly ( $s_2$ ) distributed white signals: (a) joint input pdf; (b) joint pdf of mixed signals; and (c) joint pdf of recovered signals.

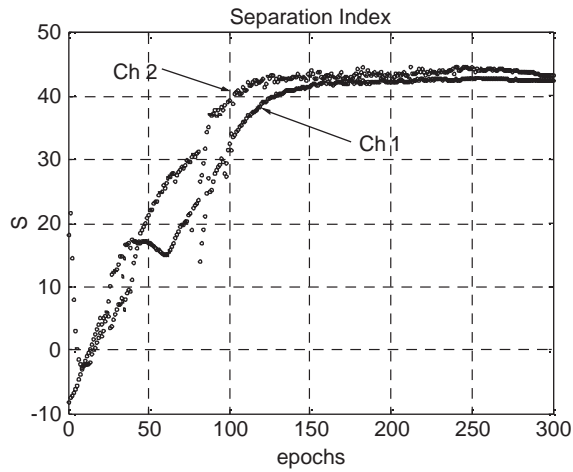


Fig. 8. Experiment 1: Separation indexes of sources during learning.

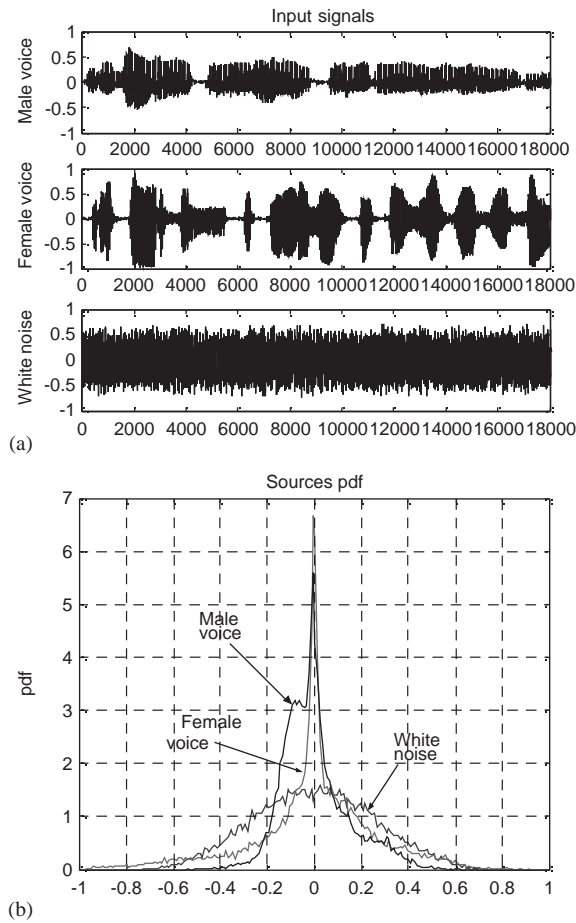


Fig. 9. Experiment 2: (a) sources considered in experiments: male speech ( $s_1$ ), female speech ( $s_2$ ) and white noise ( $s_3$ ). Sources were sampled at 8 kHz; and (b) empirical distributions of sources.

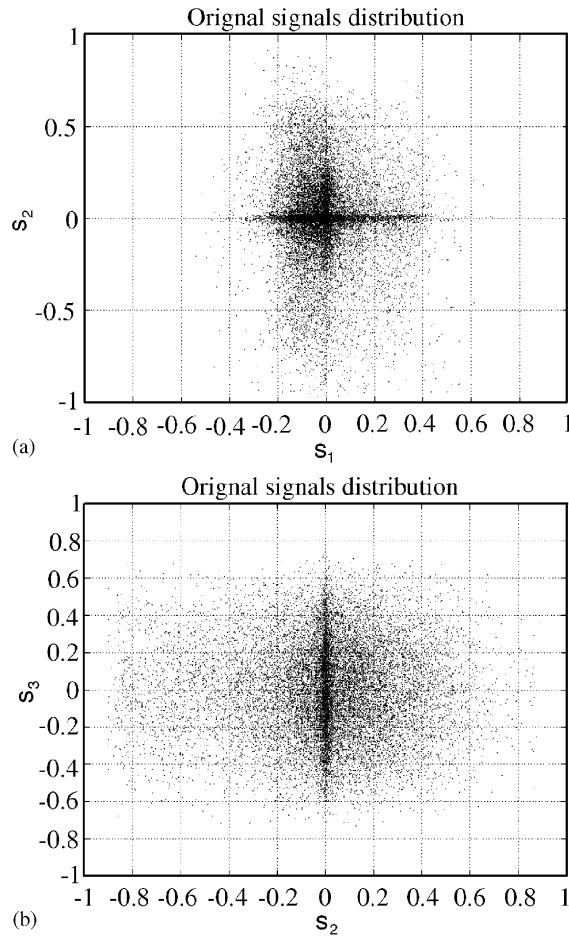


Fig. 10. Experiment 2: (a) joint pdf of signals  $[s_1, s_2]$ ; and (b) joint pdf of signals  $[s_2, s_3]$ .

For all mixing environments the nonlinear distortion applied was

$$\mathbf{F}[f_1(r_1), f_2(r_2)] = [r_1 + 2r_1^3, 0.5r_2 + \tanh(5r_2)].$$

Table 1 summarizes the experimental setup.

**Experiment 2.1.** Signals  $s_1$  and  $s_2$  were first considered. The mixing environment was

$$\mathbf{A} = \begin{bmatrix} 0.7 & -0.15 \\ 0.35 & 0.7 \end{bmatrix}, \quad \mathbf{Z}[k] = \begin{bmatrix} 0.05 - 0.025z^{-10} + 0.125z^{-30} + 0.0625z^{-30} & -0.045 - 0.035z^{-20} \\ 0.025 - 0.0125z^{-10} + 0.0645z^{-20} & 0.05 - 0.0125z^{-20} + 0.025z^{-30} \end{bmatrix}.$$

The recurrent demixing structure of Fig. 4 was employed, with 20-tap filters and 53 spline control points. The learning rates were  $\mu_{\text{spline}} = 10^{-6}$ ,  $\mu_{\text{W}} = 10^{-5}$ ,  $\mu_{\text{B}} = 10^{-5}$ . Training was stopped after 180 epochs. Fig. 11 shows the results. Effective separation was confirmed also by listening tests.

Table 1  
Experimental setup

Experiment no.	Architecture	Source pair
2.1	Recurrent	$[s_1, s_2]$
2.2	Recurrent	$[s_2, s_3]$
2.3	Feedforward	$[s_1, s_2]$
2.4	Feedforward	$[s_2, s_3]$

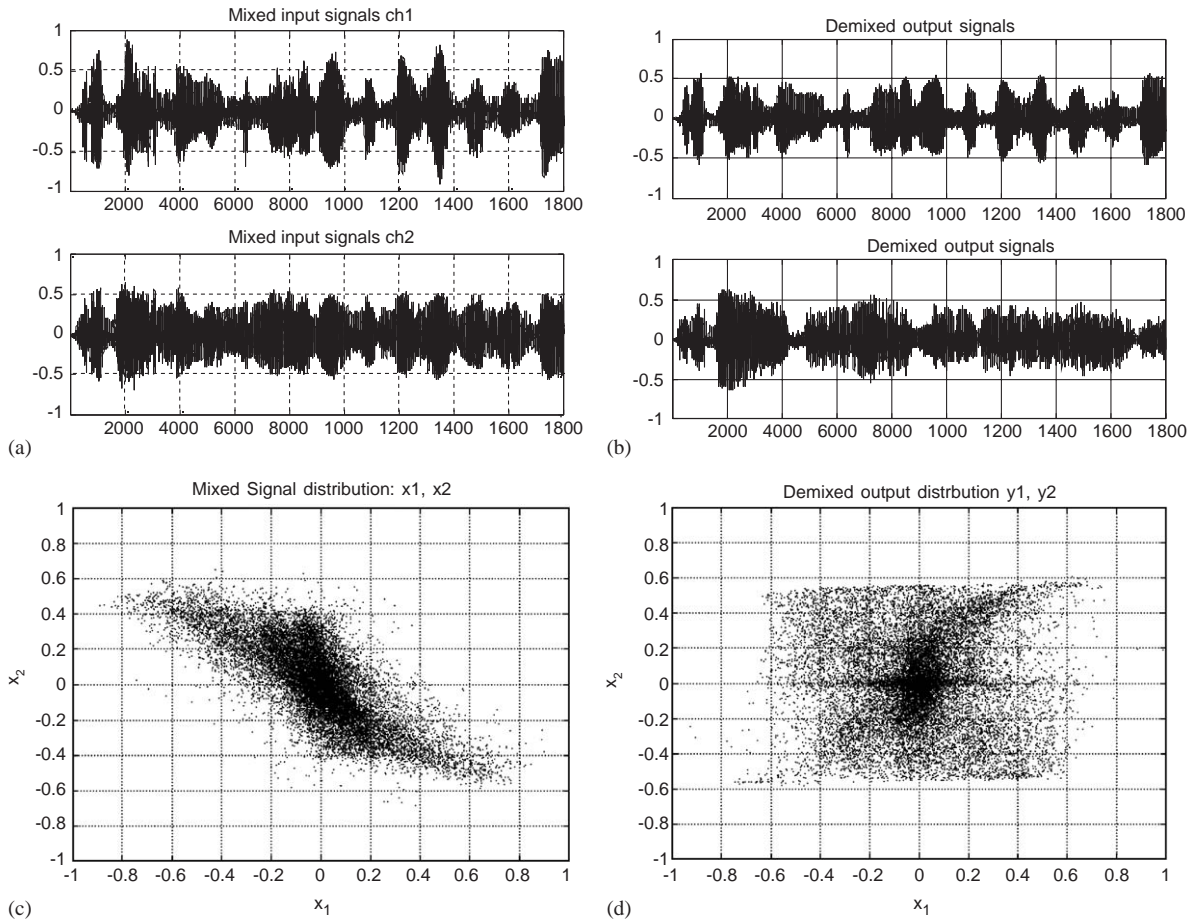


Fig. 11. Experiment 2.1: (a) mixed signals  $[x_1, x_2]$ ; (b) demixed signals  $[y_1, y_2]$ ; (c) joint pdf of  $[x_1, x_2]$ ; and (d) joint pdf of  $[y_1, y_2]$ .

Fig. 12 shows the parameters estimated by the demixing structure. It is demonstrated the capability of spline neurons in estimating both the SFs and the inverse of distorting functions.

Final estimated values of matrices  $\mathbf{P}$  and  $\mathbf{B}$  were  $\mathbf{P} = [0.801, 0.251; -0.096, 0.724]$  and  $\mathbf{B} = [0.854, 0.105; -0.085, 0.528]$ .

Finally, Fig. 13 depicts the convergence of the separation indexes during learning.



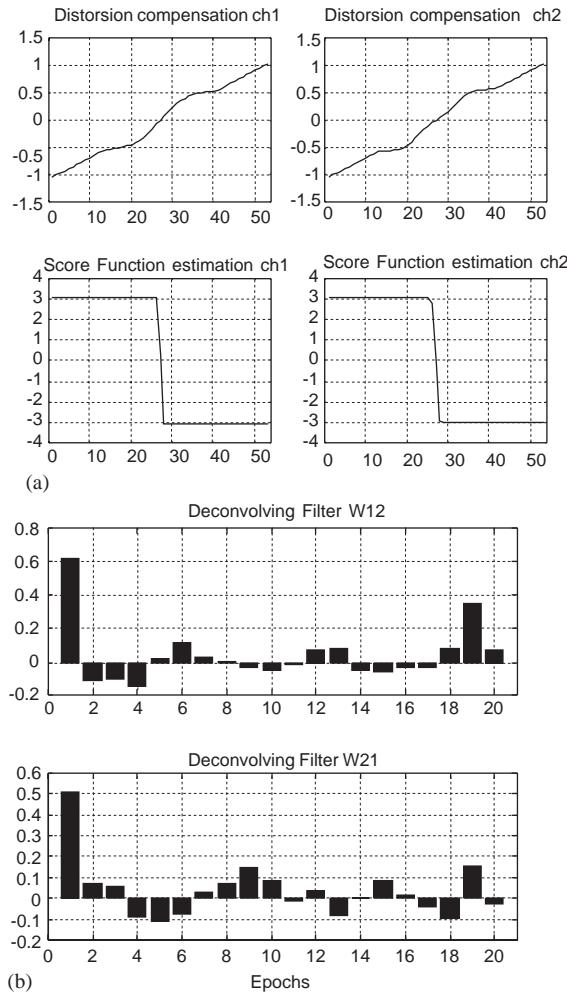


Fig. 12. Experiment 2.1: (a) estimation of the SFs and the inverse nonlinear functions; and (b) demixing filter taps for  $\mathbf{W}_{12}, \mathbf{W}_{21}$ .

**Experiment 2.2.** Signal pair  $[s_2, s_3]$  was considered in this case. The mixing matrix  $\mathbf{A}$  was of the same experiment 2.1, while  $\mathbf{Z}[k]$  was

$$\mathbf{Z}[k] = \begin{bmatrix} 0.08e^{-(k-1)/4} + 0.045e^{-(k-10)/4} & -0.03e^{-(k-1)/2} + 0.045e^{-(k-15)/4} \\ 0.04e^{-(k-1)/6} + 0.01e^{-(k-8)/6} & 0.06e^{-(k-1)/4} + 0.01e^{-(k-15)/4} \end{bmatrix}, \quad k = 1, \dots, 30.$$

The recurrent architecture was used again, with 30-tap filters and 53 spline control points. The same learning rates of Experiment 2.1 were employed. Fig. 14 shows the separation results after 80 learning epochs.

Results are quite good, especially when considering the difficult mixing environment and the reduced number of learning epochs.

Fig. 15 describes the estimation of both the SFs and the inverse distorting functions, and the behavior of the separation indexes.

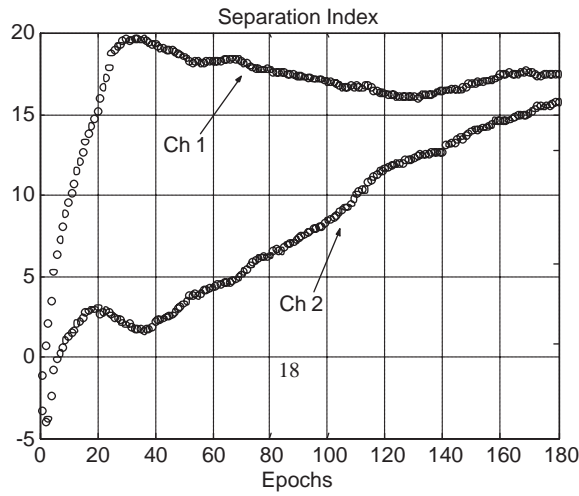


Fig. 13. Experiment 2.1: Separation indexes of sources during learning.

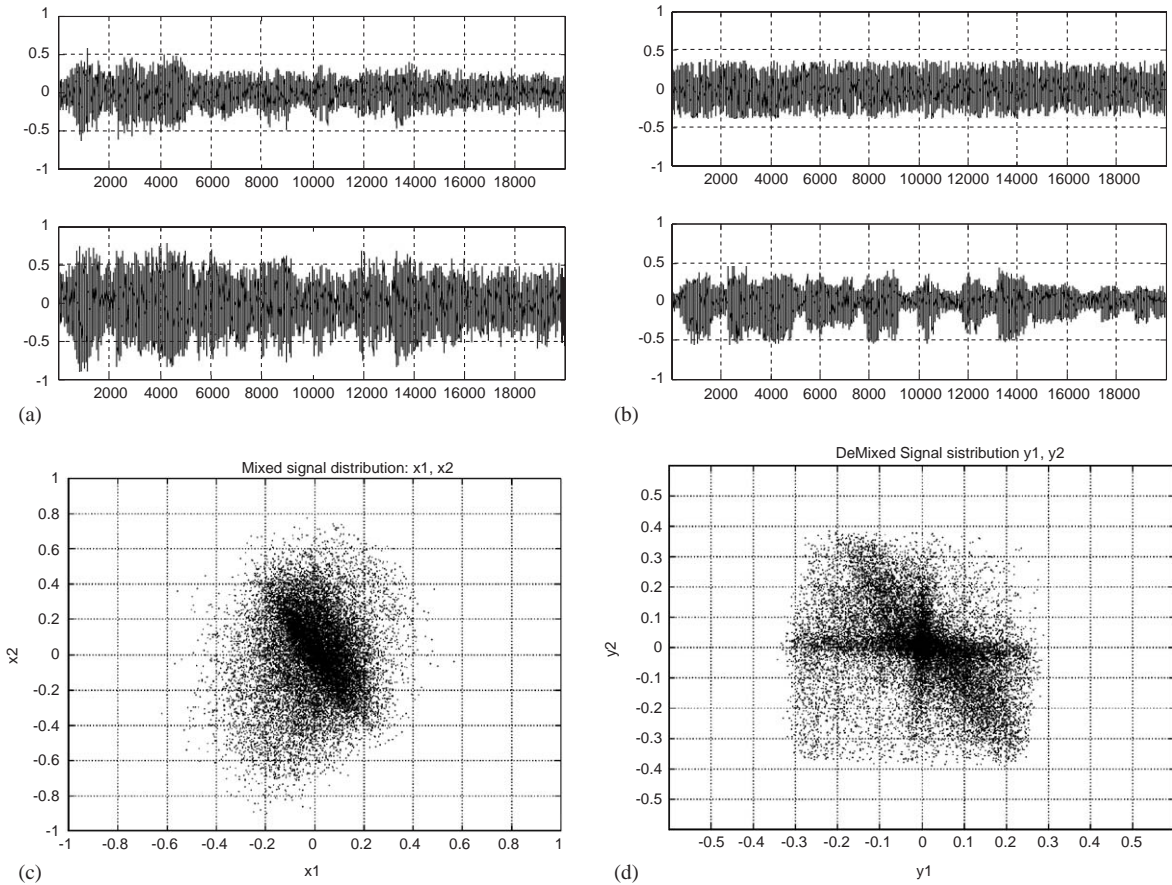


Fig. 14. Experiment 2.2: (a) mixed signals  $[x_1, x_2]$ ; (b) demixed signals  $[y_1, y_2]$ ; (c) joint pdf of  $[x_1, x_2]$ ; and (d) joint pdf of recovered signals  $[y_1, y_2]$ .

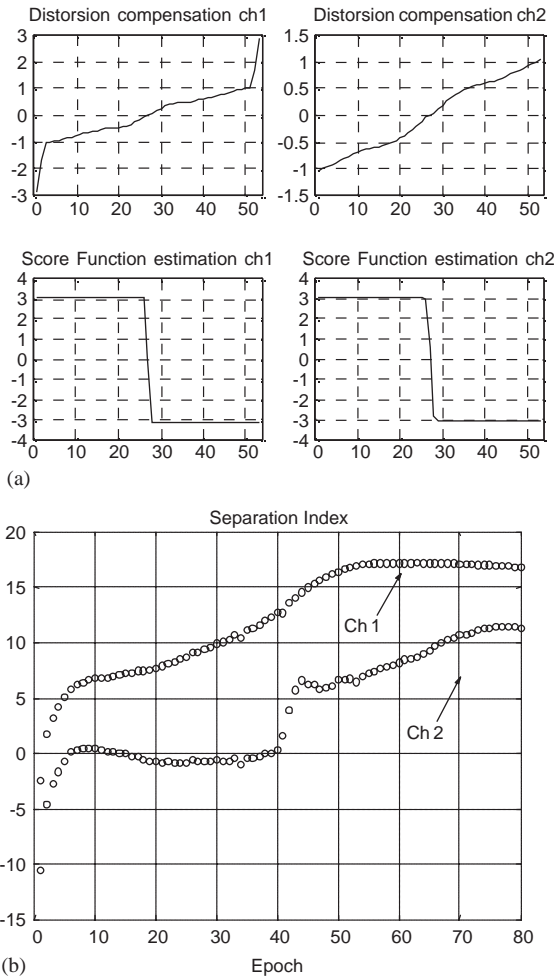


Fig. 15. Experiment 2.2: (a) estimation of the SFs and the inverse nonlinear functions; and (b) separation indexes during learning.

**Experiment 2.3.** In this experiment the signal pair  $[s_1, s_2]$  (male and female speech) was considered. The same mixing environment of Experiment 2.1 was adopted. The feedforward demixing architecture of Fig. 5 was employed, with 70-tap filters and 103 spline control points. Training was performed for 150 epochs training, with learning rates  $\mu_{\text{spline}} = 10^{-7}$ ,  $\mu_B = 7 \times 10^{-7}$ ,  $\mu_{W[k]} = 7 \times 10^{-9}$ . Fig. 16 shows the recovered signals and their joint distribution.

Fig. 17 shows the estimation of the SFs and of the inverse of distorting functions, and the separation indexes.

**Experiment 2.4.** Mixing of signals  $[s_2, s_3]$  was again considered, with the same mixing environment of Experiment 2.2. The feedforward architecture was applied, with 70-tap filters and 103 spline control points. One hundred and fifty training epochs and the same learning rates of previous experiment were considered.

Recovered signals and their joint distribution are presented in Fig. 18. Fig. 19(a) shows the filter taps of the demixing structure  $\mathbf{W}[k]$ , while Fig. 19(b) depicts the evolution of the separation indexes.

Finally, matrix  $\mathbf{B}$  was estimated as  $\mathbf{B} = [0.898, 0.202; 0.329, 0.693]$ .

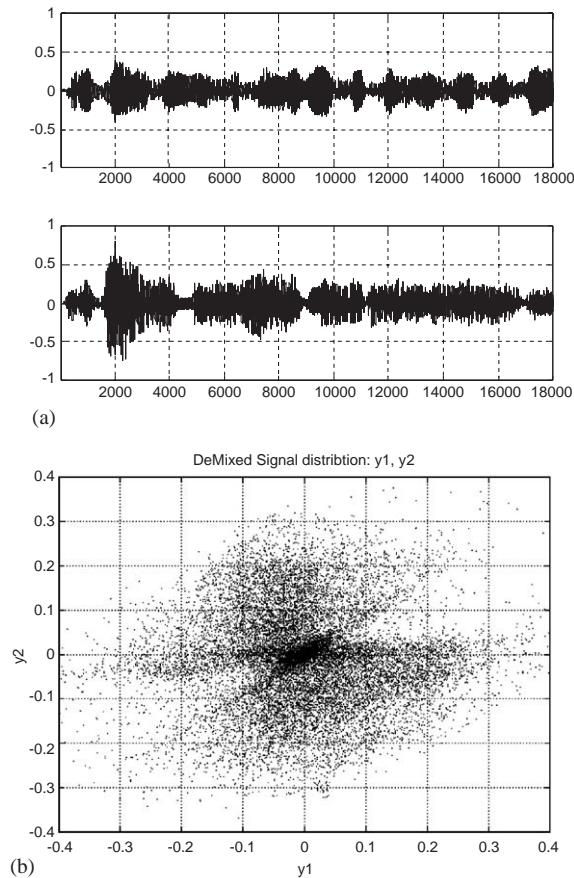


Fig. 16. Experiment 2.3: (a) demixed signals  $[y_1, y_2]$ ; and (b) joint pdf of recovered signals  $[y_1, y_2]$ .

**Remarks.** A first consideration is that the recurrent architectures produce demixed outputs with the same quality of the FIR networks but using a reduced set of learning parameters, with the consequence of a lower computational cost. The number of spline control points does not substantially affect the computational effort, since each learning step involves only four control points. A critical issue concerning the choice of the spline control points is the occurrence of low-value outputs during the learning phase, with the risk of a low estimation resolution. Convergence of the algorithms was demonstrated by averaging through an extensive set of different environments, for each pair of sources and each recovering architecture. For each set of sources, 20 experiments were performed for both architectures, with different initializations, and the average separation index was computed. Table 2 summarizes these results.

The average values of the separation index in Table 2 provide an empirical verification of the algorithm convergence also when correlated sources are considered.

## 5. Conclusion

In this paper a novel type of mixing environment consisting in the convolutive mixing of PNL mixtures was introduced and described. Proof of existence and uniqueness of the solution was provided. The

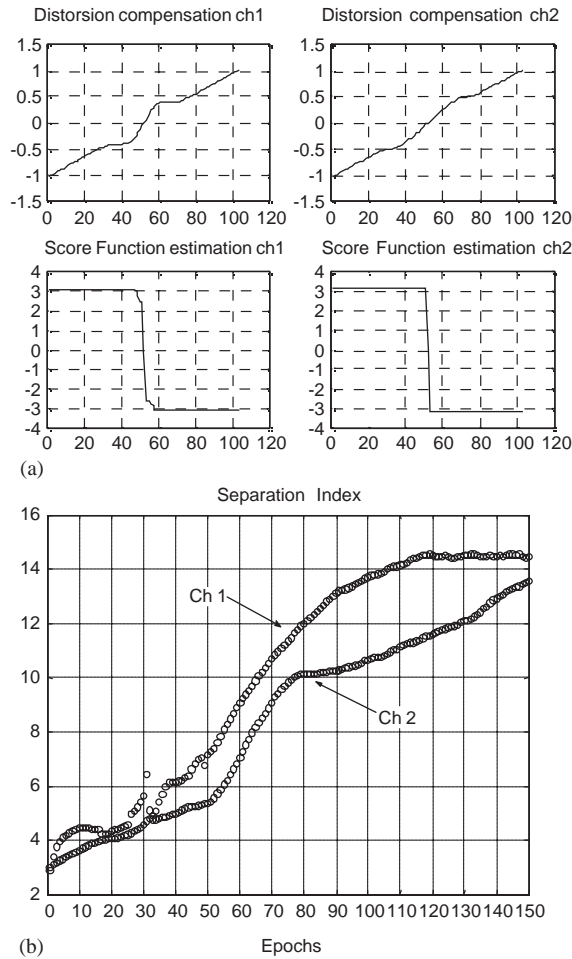


Fig. 17. Experiment 2.3: (a) estimation of the SFs and of the inverse of nonlinear functions; and (b) separation indexes.

proposed separation algorithms employed flexible spline networks to perform local on-line estimation of the unknown functions (pdf and nonlinear compensating functions).

Since the natural gradient was used, the learning algorithm of every *single* neural block is equivariant by itself, but considering all the three blocks together this attractive property cannot be guaranteed, thus possibly leading to an increase of the learning time. Quality of the results was verified in different mixing environmental conditions in terms of separation indexes of the sources.

The proposed algorithm works by enforcing both spatial and temporal independence by minimizing the cost function proposed in (8).

The best results have been obtained with white sources but good performances have been reached also in the separation of voice signals (that are not white) due to the robustness of the proposed algorithm.

### Appendix A. Proof of Proposition 1

*Sufficient condition:* existence of the solution.

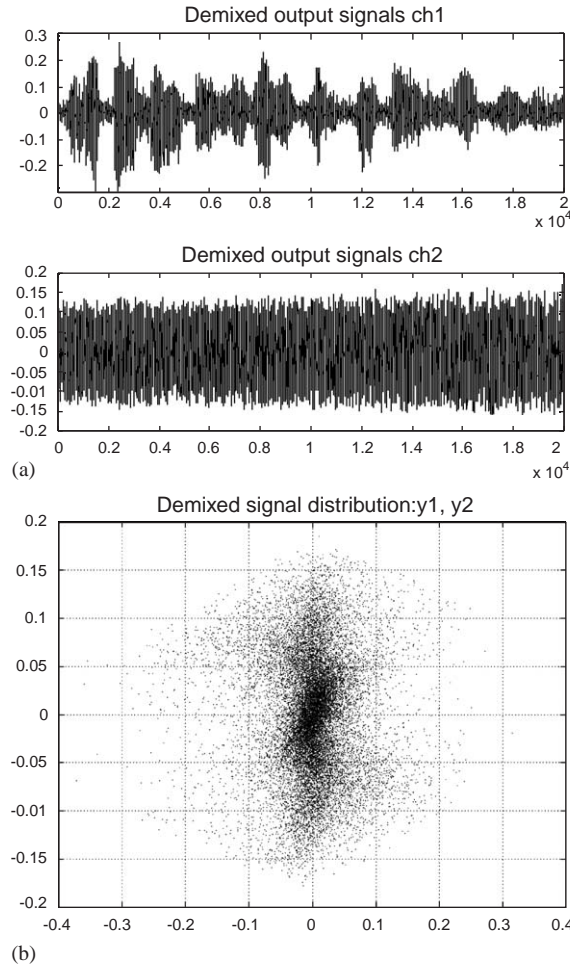


Fig. 18. Experiment 2.4: (a) demixed signals  $[y_1, y_2]$ ; and (b) joint pdf of  $[y_1, y_2]$ .

Given the channel model (7), it is easy to verify that if  $\mathbf{s}[n]$  is a spatially independent random vector, under the given assumptions,  $\mathbf{y}[n]$  will be spatially independent too and the channel does not produce any mixing. Given the mixing model  $\mathcal{F}\{\mathbf{A}, \mathbf{F}, \mathbf{Z}\}$ , assumptions (a)–(c) guarantee that there exist a matrix  $\mathbf{B}$ ,  $N$  functions  $g_i[\bullet]$  and a matrix  $\mathbf{W}[k]$  in which each element is an MA filter of at most infinite order,<sup>5</sup> such that:

$$\begin{aligned}
 \mathbf{Z}[k] * \mathbf{W}[k] &= \mathbf{I}, \\
 \mathbf{G}\{\mathbf{F}\{\mathbf{A}\mathbf{s}[n]\}\} &= \mathbf{A}\mathbf{s}[n], \\
 \mathbf{B}\mathbf{A}\mathbf{s}[n] &= \mathbf{s}[n].
 \end{aligned} \tag{35}$$

Based on (35), the input–output transformation can be written as  $\mathbf{B}\mathbf{G}\{\mathbf{W}[n] * \mathbf{Z}[n]\mathbf{F}\{\mathbf{A}\mathbf{s}[n]\}\} = \mathbf{P}\mathbf{A}\mathbf{D}\mathbf{s}[n]$ .

*Necessary condition:* uniqueness of the solution.

<sup>5</sup>An important corollary of the famous Wold decomposition [50] ensures that an ARMA model can be represented as MA model or AR model of at most infinite order. In practice, finite order MA models for those infinite order are often sufficiently accurate, because the order of the filter was properly chosen to yield acceptable error in inversion.

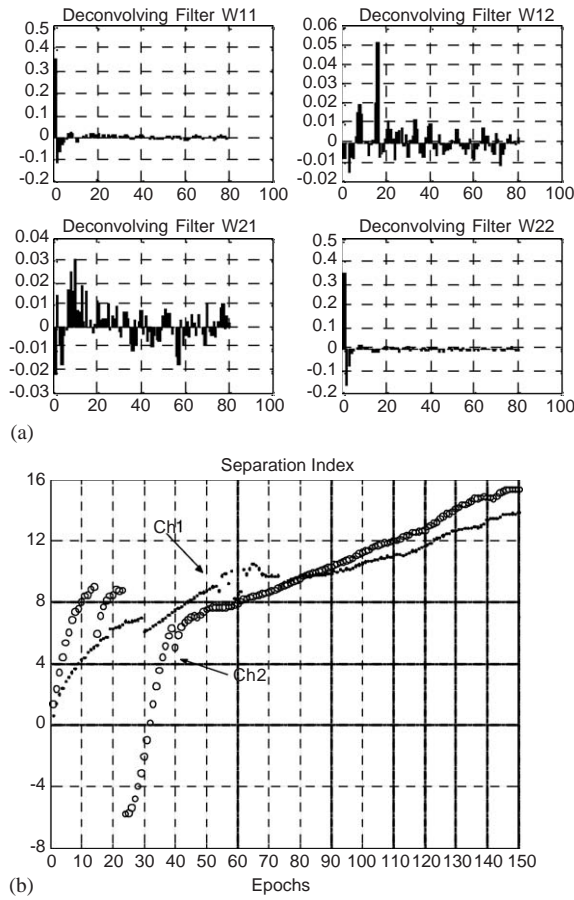


Fig. 19. Experiment 2.4: (a) demixing filter taps  $w_{11}$ ,  $w_{12}$ ,  $w_{21}$ ,  $w_{22}$ ; and (b) separation indexes during learning.

Table 2  
Average performance of the proposed algorithms

Original sources	Used network	Mean $S$ (dB) [Ch1, Ch2]
Set 1 [ $s_1, s_2$ ] Male & female speech	Feedforward	[13.90; 14.83]
	Recurrent	[15.19; 14.64]
Set 2 [ $s_2, s_3$ ] Female speech & white noise	Feedforward	[15.59; 14.10]
	Recurrent	[13.21; 14.05]

This is to prove that if  $y[n]$  is a spatially independent random vector, the channel model must be (7). The transformation mapping  $s$  into  $y$  is

$$y[n] = \mathbf{BG} \left[ \sum_{k=0}^{L_W-1} \mathbf{W}[k] \sum_{p=0}^{L_Z-1} \mathbf{Z}[k] \mathbf{F}\{\mathbf{A}s[n-k-p]\} \right]$$

$$= \mathbf{B} \begin{bmatrix} g_1 \left( \sum_{j=1}^N h_{1j}[k] * f_j \left( \sum_{i=1}^N a_{ji} s_i[n] \right) \right) \\ \vdots \\ g_N \left( \sum_{j=1}^N h_{Nj}[k] * f_j \left( \sum_{i=1}^N a_{ji} s_i[n] \right) \right) \end{bmatrix}. \tag{36}$$

Given the channel  $\mathbf{H}[k] = \mathbf{W}[k] * \mathbf{Z}[k]$  (using the notation of Lambert [26]), let  $\mathcal{B}$  be the set of pairs of indexes  $(i, j)$  such that  $h_{ij}[k] \neq 0$  for some integer  $k$ . We first introduce the two following propositions.

**Proposition A.** Under the assumptions of Proposition 1, for each possible pair  $(i, j) \in \mathcal{B}$  there is at most one value  $k = \bar{k}$  such that  $h_{ij}[\bar{k}] \neq 0$ . Moreover considering two different pairs  $(i, j)$  and  $(k, l)$  in  $\mathcal{B}$  then  $\bar{k}_{ij} = \bar{k}_{kl}$ .

**Proof.** Considering that both  $\mathbf{B}$ ,  $\mathbf{A}$ ,  $\mathbf{F}()$  and  $\mathbf{G}()$  are static applications but  $\mathbf{H}$  is convolutive so, if there were more than one value  $\bar{k}$ , the output signal  $\mathbf{y}[n]$  could not be white as assumed in hypothesis. Indeed considering the second part of Eq. (36) for the generic  $i$ th output channel:  $y_i[n] = e_i(s_1[n], s_1[n - 1], \dots, s_1[n - L_H], \dots, s_N[n], \dots, s_N[n - L_H])$ ; then it follows that the temporal correlation for the same output channel is:  $E\{y_i[n], y_i[n - L]\} \neq 0$  when  $0 < L \leq L_H$ .

If different pairs  $(i, j)$  and  $(k, l)$  of  $\mathcal{B}$  were such that  $\bar{k}_{ij} \neq \bar{k}_{kl}$ , then the recovery structure (6), assumed for hypothesis, would not be able anymore to provide independent white outputs considering that both  $\mathbf{B}$ ,  $\mathbf{A}$ ,  $\mathbf{F}()$  and  $\mathbf{G}()$  are static applications:  $y_i[n] = e_i(s_1[n - \bar{k}_1], s_2[n - \bar{k}_1], \dots, s_N[n - \bar{k}_1], \dots, s_1[n - \bar{k}_N], \dots, s_N[n - \bar{k}_N])$  (in which  $\bar{k}_1, \dots, \bar{k}_N$  are  $N$  possible values for  $k$ ).  $\square$

**Proposition B.** Under the assumptions of Proposition 1, for a given  $k = \bar{k}$  verifying Proposition A, in matrix  $\mathbf{H}[\bar{k}]$  there are no pairs of indexes  $(i, j) \in \mathcal{B}$  having the same  $i$  or  $j$  components, moreover the cardinality of  $\mathcal{B}$  cannot be smaller than the number of columns (or rows). In other words  $(i, j), (h, m) \in \mathcal{B} \Rightarrow i \neq h, j \neq m$ .

**Proof.** The proof is based on geometrical consideration about the support of joint pdf; let us consider the global input–output model (36) represented in Fig. 20.

It is well known that  $N$  independent signals (of finite support, as hypothesis (d) of Proposition 1 assumes) have joint pdf in an “hyper rectangular” support (see [51] for problem in  $\mathbb{R}^2$ , an example is in Fig. 21(a)) so  $\mathbf{s} \in I_S : [a_1, b_1] \times \dots \times [a_N, b_N]$ ; the same can be assumed for the independent outputs so:  $\mathbf{y} \in I_y : [a'_1, b'_1] \times \dots \times [a'_N, b'_N]$ .

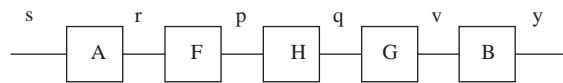


Fig. 20. Input–output global model.

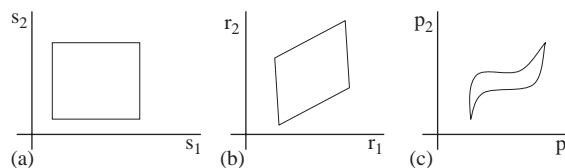


Fig. 21. Joint distribution “borders” in  $\mathbf{s}$  plane (a),  $\mathbf{r}$  plane (b) and  $\mathbf{p}$  plane (c).



It is well known that the  $\mathbb{R}^2$  linear mapping  $\mathbf{M} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}$  of an independent  $\mathbb{R}^2$  random vector produces a random vector for which the support of the joint pdf is a parallelogram where the slope of the borders are  $\alpha_{11}/\alpha_{21}$  and  $\alpha_{21}/\alpha_{22}$  as in Fig. 21(b). Extending the concept to  $\mathbb{R}^N$ , the vector  $\mathbf{r} : \mathbf{r} = \mathbf{A}\mathbf{s}$  has a joint distribution with hyper-parallelogram support on the  $\mathbf{r}$  space. The vector  $\mathbf{p}$

$$\mathbf{p} : \begin{cases} p_1 = f_1(r_1) \\ \vdots \\ p_N = f_N(r_N) \end{cases} \tag{37}$$

has the pdf with the typical support of ‘‘PNL mixing’’; an example in  $\mathbb{R}^2$  is in Fig. 21(b).

For the hypothesis of output independence  $\mathbf{y}$  has the same hyper-rectangular support of the vector  $\mathbf{s}$  admitting at most a different scaling factor. If  $b_{ij}^{-1}$  is the  $ij$ th element of the inverse of the matrix  $\mathbf{B}$ , the vector

$$\mathbf{v} : \begin{cases} v_1 = b_{11}^{-1}y_1 + b_{12}^{-1}y_2 + \dots + b_{1N}^{-1}y_N \\ \vdots \\ v_N = b_{N1}^{-1}y_1 + b_{N2}^{-1}y_2 + \dots + b_{NN}^{-1}y_N \end{cases} \tag{38}$$

has an hyper-parallelogram domain generally different from the one of vector  $\mathbf{r}$ . Moreover the vector  $\mathbf{r}$  differs from the vector  $\mathbf{v}$  for at most a linear nonsingular transform :  $\exists \mathbf{C} : \mathbf{r} \longrightarrow \mathbf{v}$ .

$$\mathbf{v} = \begin{bmatrix} c_{11} & \dots & c_{1N} \\ \vdots & \ddots & \vdots \\ c_{N1} & \dots & c_{NN} \end{bmatrix} \mathbf{r}. \tag{39}$$

Considering  $g_i^{-1}(v_i)$  (the inverse of  $i$ th function  $g_i(\cdot)$  that always exists by assumptions), the vector:

$$\mathbf{q} : \begin{cases} q_1 = g_1^{-1}(v_1) \\ \vdots \\ q_N = g_N^{-1}(v_N) \end{cases} \tag{40}$$

has the pdf with the typical support of ‘‘PNL mixing’’; an example in  $\mathbb{R}^2$  is in Fig. 21(c). From the structure of the global input–output model (36)<sub>H</sub> (Fig. (20)) it is possible to see that vector  $\mathbf{q}$  can be obtained from vector  $\mathbf{p}$  by the linear mapping  $\mathbf{H} : \mathbf{p} \longrightarrow \mathbf{q}$ .

$$\mathbf{q} = \mathbf{H}\mathbf{p}. \tag{41}$$

Finally (39) and (41) must both hold. By substitution of (40), (41) and (37) inside (39), it results:

$$\begin{aligned} c_{11}r_1 + c_{12}r_2 + \dots + c_{1N}r_N &= g_1(h_{11}f_1(r_1) + h_{12}f_2(r_2) + \dots + h_{1N}f_N(r_N)) \\ \vdots & \\ c_{N1}r_1 + c_{N2}r_2 + \dots + c_{NN}r_N &= g_N(h_{N1}f_1(r_1) + h_{N2}f_2(r_2) + \dots + h_{NN}f_N(r_N)). \end{aligned} \tag{42}$$

Considering the well-known results about the uniqueness of the inverse function, (42) is true if and only if at least one of the following is true:

1. nonlinear function  $g_i$  and  $f_i$ , for each  $i = 1, \dots, N$  are in effect linear functions;
2. for the right side of each equation of (42)  $\forall i, \exists !j : h_{ij} \neq 0$ .

The first assumption is in contrast with the hypothesis. So if more than one pair  $(i, j) \in \mathcal{B}$  had the same first (or second) component, at least two output variables would not be independent considering the structure of the input output model (5) and (6).

Then if the cardinality of  $\mathcal{B}$  were less than the number of rows (or columns) it would not be possible to obtain  $N$  independent output components.  $\square$

Merging results of Propositions A and B, the pdf of  $\mathbf{s}$  can be written as a function of the pdf of  $\mathbf{y}$

$$p_{\mathbf{s}}(\mathbf{s}) = \prod_{i=1}^N p_{s_i}(s_i) = \prod_{i=1}^N p_{y_i} \left( \sum_{j=1}^N b_{ij} g_j \left[ h_{jq}[k] f_q \left( \sum_{m=1}^N a_{qm} s_m[n-k] \right) \right] \right) |\mathbf{J}| \quad \forall \mathbf{s} \in \mathbb{R}^N. \tag{43}$$

Proposition B grants that in (43)  $\forall j \in 1, \dots, N, \exists! q, k : h_{jq}[k] \neq 0$ .

From assumption (c)  $\exists \bar{\mathbf{s}} \in \mathbb{R}^N$  such that  $p_{\mathbf{s}}(\bar{\mathbf{s}}) \equiv 0$ . From (36), for a nonnull Jacobian  $\mathbf{J}$ , there exists some  $\mathbf{y}^0 = [y_1^0, \dots, y_N^0] \in \mathbb{R}^N$  such that  $\prod_{i=1}^N p_{y_i}(y_i^0) = 0$ . Consequently there is at least one integer  $i$  such that  $p_{y_i}(y_i^0) = 0$ . This leads to the following equation:

$$y_i^0 = \sum_{j=1}^N b_{ij} g_j \left[ h_{jq}[k] f_q \left( \sum_{m=1}^N a_{qm} s_m[n-k] \right) \right]. \tag{44}$$

Solutions of (44) lie on  $\mathcal{H}_i(\mathbf{s})$ , which is a hypersurface in  $\mathbb{R}^N$ . It is evident that  $\forall \mathbf{s} \in \mathcal{H}_i(\mathbf{s}) \Rightarrow p_{\mathbf{s}}(\mathbf{s}) = 0$ . For a given  $i$ ,  $\mathcal{H}_i(\mathbf{s})$  is parallel to the hyperplane orthogonal to the axis  $s_i$ .

Suppose that  $\mathcal{H}_i(\mathbf{s})$  is not parallel to any  $s_i = 0$  plane. The projection of  $\mathcal{H}_i(\mathbf{s})$  onto  $s_i$  should be  $\mathbb{R} : \forall s_i \in \mathbb{R} \exists s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_N : \mathbf{s} \in \mathcal{H}_i \Rightarrow p_{\mathbf{s}}(\mathbf{s}) \equiv 0$ . This cannot be true since  $\int_{\mathcal{S}} p_{\mathbf{s}}(\mathbf{s}) d\mathbf{s} = 1$ . Without loss of generality, it should be noted that:

$$\sum_{j=1}^N b_{ij} g_j \left[ h_{jq}[k] f_q \left( \sum_{m=1}^N a_{qm} s_m[n-k] \right) \right] = w_{\sigma(i)}(s_{\sigma(i)}), \quad i = 1, \dots, N, \tag{45}$$

where  $w_{\sigma(i)}(s_{\sigma(i)})$  is a generic function depending only on  $s_{\sigma(i)}$  (that is the source for the  $i$ th output). Considering the results of Propositions A and B, (44) can be written for each channel

$$\begin{aligned} y_i &= \sum_{j=1}^N b_{ij} g_j [h_{jq}[k] f_q((\mathbf{A})_q \mathbf{s})], \\ y_{\bar{i}} &= \sum_{j=1}^N b_{\bar{i}j} g_j [h_{jp}[k] f_p((\mathbf{A})_p \mathbf{s})], \end{aligned} \tag{46}$$

where  $(\mathbf{A})_q$  is the  $q$ th row of matrix  $\mathbf{A}$ . Then without any loss of generality taking  $\sigma(i) = i$

$$\sum_{j=1}^N b_{ij} g_j \left[ h_{jq}[k] f_q \left( \sum_{m=1}^N a_{qm} s_m[n-k] \right) \right] = w_i(s_i), \quad i = 1, \dots, N. \tag{47}$$

Derivation with respect to  $\mathbf{s}$  yields

$$\begin{bmatrix} \dot{w}_1(s_1) & 0 \\ 0 & \dot{w}_N(s_N) \end{bmatrix} = \mathbf{B} \begin{bmatrix} \dot{g}_1 \left[ h_{1q}[k] f_q \left( \sum_{m=1}^N a_{qm} s_m [n-k] \right) \right] & 0 \\ 0 & \dot{g}_N[\mathbf{s}] \end{bmatrix} \\ \times \begin{bmatrix} h_{11}[k] & 0 \\ 0 & h_{NN}[k] \end{bmatrix} \begin{bmatrix} \dot{f}_1 \left( \sum_{m=1}^N a_{1m} s_m [n-k] \right) & 0 \\ 0 & \dot{f}_1(\mathbf{s}) \end{bmatrix} \mathbf{A}. \quad (48)$$

Considering  $\mathbf{s}_1$  and  $\mathbf{s}_2$  as coordinates of the hypersurface  $\mathcal{H}(s)$ , (48) should be evaluated in  $\mathbf{s}_1$  and in  $\mathbf{s}_2$  as follows:

$$\begin{cases} \mathbf{D}(\mathbf{s}_1) = \mathbf{B} \Lambda_{\dot{G}}(\mathbf{s}_1) \mathbf{H} \Lambda_{\dot{F}}(\mathbf{s}_1) \mathbf{A} \\ \mathbf{D}(\mathbf{s}_2) = \mathbf{B} \Lambda_{\dot{G}}(\mathbf{s}_2) \mathbf{H} \Lambda_{\dot{F}}(\mathbf{s}_2) \mathbf{A} \end{cases} \rightarrow \begin{cases} \mathbf{D}(\mathbf{s}_1) = \mathbf{B} \Lambda_{\dot{G}\dot{H}\dot{F}}(\mathbf{s}_1) \mathbf{A} \\ \mathbf{D}(\mathbf{s}_2) = \mathbf{B} \Lambda_{\dot{G}\dot{H}\dot{F}}(\mathbf{s}_2) \mathbf{A}. \end{cases} \quad (49)$$

Then eliminating  $\mathbf{B}$ :

$$\mathbf{A} \underbrace{\mathbf{D}^{-1}(\mathbf{s}_2) \mathbf{D}(\mathbf{s}_1)} = \underbrace{\Lambda_{\dot{G}\dot{H}\dot{F}}^{-1}(\mathbf{s}_2) \Lambda_{\dot{G}\dot{H}\dot{F}}(\mathbf{s}_1)} \mathbf{A}.$$

$$\begin{bmatrix} d_{11}(\mathbf{s}_2, \mathbf{s}_1) & & 0 \\ & \ddots & \\ 0 & & d_{NN}(\mathbf{s}_2, \mathbf{s}_1) \end{bmatrix} = \begin{bmatrix} \lambda_{11}(\mathbf{s}_2, \mathbf{s}_1) & & 0 \\ & \ddots & \\ 0 & & \lambda_{NN}(\mathbf{s}_2, \mathbf{s}_1) \end{bmatrix}$$

As expressed in assumption (a), matrix  $\mathbf{A}$  is regular. For each pair of nonzero elements of  $\mathbf{A}$  it is possible to write:

$$\begin{cases} a_{ij}[d_{jj}(\mathbf{s}_2, \mathbf{s}_1) - \lambda_{ii}(\mathbf{s}_2, \mathbf{s}_1)] = 0 \\ a_{hj}[d_{jj}(\mathbf{s}_2, \mathbf{s}_1) - \lambda_{hh}(\mathbf{s}_2, \mathbf{s}_1)] = 0 \end{cases} \Rightarrow \lambda_{ii}(\mathbf{s}_2, \mathbf{s}_1) = \lambda_{hh}(\mathbf{s}_2, \mathbf{s}_1) \quad \forall \mathbf{s}_2, \mathbf{s}_1 \in \mathcal{H}. \quad (50)$$

From (50) it follows:

$$\frac{\dot{g}_i \tilde{f}_{\sigma(i)}((\mathbf{A})_{\sigma(i)} \mathbf{s}_1) \tilde{f}_{\sigma(i)}((\mathbf{A})_{\sigma(i)} \mathbf{s}_1)}{\dot{g}_h \tilde{f}_{\sigma(h)}((\mathbf{A})_{\sigma(h)} \mathbf{s}_2) \tilde{f}_{\sigma(h)}((\mathbf{A})_{\sigma(h)} \mathbf{s}_2)} = C \quad \forall \mathbf{s}_2, \mathbf{s}_1, \quad (51)$$

where  $\tilde{f}_{\sigma(i)}((\mathbf{A})_{\sigma(i)} \mathbf{s}_1) = \alpha f_{\sigma(i)}((\mathbf{A})_{\sigma(i)} \mathbf{s}_1)$  and  $C$  is a constant. For the two linear forms in (50)  $(\mathbf{A})_{\sigma(i)} \mathbf{s}_1$   $(\mathbf{A})_{\sigma(h)} \mathbf{s}_2$  are independent as assumed in (a), it is possible to express (50) in the following way:  $\dot{g}_i \tilde{f}_{\sigma(i)}(x) \tilde{f}_{\sigma(i)}(x) = C \dot{g}_h \tilde{f}_{\sigma(h)}(y) \tilde{f}_{\sigma(h)}(y) \quad \forall x, y \in \mathbb{R}$ . This can be true if and only if  $g_i(\cdot)$  is the inverse of  $f_{\sigma(i)}(\cdot)$ , up to a scaling factor.

From (36), by previous results it follows:

$$\mathbf{y}[n] = \mathbf{B} \begin{bmatrix} \xi_1 & & \\ & \ddots & \\ & & \xi_N \end{bmatrix} \mathbf{A} \mathbf{s}[n-k], \quad (52)$$

where  $\xi_1, \dots, \xi_N$  are scaling coefficients.

This method above reduces the mixing–demixing nonlinear convolutive channel (A2) to the simpler, linear and static model (52). By considering

$$\tilde{\mathbf{B}} = \mathbf{B} \begin{bmatrix} \xi_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \xi_N \end{bmatrix} \tag{53}$$

and  $\hat{\mathbf{s}}[n] = \mathbf{s}[n - k]$ , (52) turns into  $\mathbf{y} = \tilde{\mathbf{B}}\mathbf{A}\hat{\mathbf{s}}$ . For this formulation independent outputs can be obtained if and only if

$$\tilde{\mathbf{B}}\mathbf{A} = \mathbf{P} \begin{bmatrix} \varepsilon_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \varepsilon_N \end{bmatrix}, \tag{54}$$

where  $\mathbf{P}$  is a permutation matrix.  $\square$

**Appendix B**

The Jacobian matrix in (21) has been derived extending the method used in [11] to the novel mixing–demixing environment introduced in this paper. Considering vectors  $\mathcal{Y} = [\mathbf{y}[0], \dots, \mathbf{y}[M]]^T$  and  $\mathcal{X} = [\mathbf{x}[0], \dots, \mathbf{x}[M]]^T$ , it is possible to express the demixing model as

$$\mathcal{Y} = \mathcal{B}\mathcal{G}(\mathcal{W}\mathcal{X}), \tag{55}$$

in which:

$$\mathcal{W} = \begin{pmatrix} \mathbf{W}[0] & 0 & \dots & \dots & 0 \\ \mathbf{W}[1] & \mathbf{W}[0] & 0 & & \vdots \\ \vdots & \mathbf{W}[1] & \mathbf{W}[0] & 0 & \vdots \\ \mathbf{W}[L_W] & & \ddots & \mathbf{W}[0] & 0 \\ 0 & \mathbf{W}[L_W] & \dots & \mathbf{W}[1] & \mathbf{W}[0] \end{pmatrix}, \quad \mathcal{G}[\mathbf{v}] = \begin{bmatrix} \mathbf{G}(\mathbf{v}[1]) \\ \vdots \\ \mathbf{G}(\mathbf{v}[M]) \end{bmatrix},$$

$$\mathbf{G}(\mathbf{v}[1]) = \begin{bmatrix} g(v_1[1]) \\ \vdots \\ g(v_N[1]) \end{bmatrix}$$

and

$$\mathcal{B} = \begin{pmatrix} \mathbf{B} & 0 & \dots & \dots & 0 \\ 0 & \mathbf{B} & 0 & & \vdots \\ \vdots & 0 & \mathbf{B} & 0 & \vdots \\ \vdots & & 0 & \mathbf{B} & 0 \\ 0 & \dots & \dots & 0 & \mathbf{B} \end{pmatrix}.$$

The Jacobian of (55) is

$$\begin{aligned}
 \det(J) &= \det \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_M} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial y_M}{\partial x_1} & \dots & \dots & \frac{\partial y_M}{\partial x_M} \end{pmatrix} \\
 &= \det \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & 0 & \dots & 0 \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & 0 & \vdots \\ \vdots & & \ddots & 0 \\ \frac{\partial y_M}{\partial x_1} & \dots & \dots & \frac{\partial y_M}{\partial x_M} \end{pmatrix} \\
 &= \prod_{i=1}^M \det \left( \frac{\partial y_i}{\partial x_i} \right). \tag{56}
 \end{aligned}$$

The global Jacobian matrix is a block triangular matrix, each block of the diagonal of (56) is defined as

$$\frac{\partial y_h}{\partial x_h} = J^{\text{small}}[h] = \det \begin{pmatrix} \frac{\partial y_1[h]}{\partial x_1[h]} & \frac{\partial y_1[h]}{\partial x_2[h]} & \dots & \frac{\partial y_1[h]}{\partial x_N[h]} \\ \frac{\partial y_2[h]}{\partial x_1[h]} & \frac{\partial y_2[h]}{\partial x_2[h]} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial y_N[h]}{\partial x_1[h]} & \dots & \dots & \frac{\partial y_N[h]}{\partial x_N[h]} \end{pmatrix} \tag{57}$$

in which

$$\begin{aligned}
 J_{ij}^{\text{small}}[h] &= \frac{\partial y_i[h]}{\partial x_j[h]} = \sum_{m=1}^N \left( b_{im} \dot{g}_m(v_m) \left( \frac{\partial}{\partial x_j[h]} \sum_{k=0}^{L_i} \sum_{p=1}^N w_{mp}[k] x_p[h-k] \right) \right) \\
 &= \sum_{m=1}^N (b_{im} \dot{g}_m(v_m) w_{mj}[0] x_j). \tag{58}
 \end{aligned}$$

For a better understanding it is possible to develop the computation of (58) in  $\mathbb{R}^2$  and then extend it to  $\mathbb{R}^N$ . The expression of (58) in  $\mathbb{R}^2$  is

$$J_{ij}^{\text{small}} = \sum_{M=1}^2 b_{im} \dot{g}_m[v_m] w_{mj}[0].$$

The determinant of one block is

$$\det(J^{\text{small}}) = \left( \sum_{m=1}^2 b_{1m} \dot{g}_m[v_m] w_{m1}[0] \sum_{m=1}^2 b_{2m} \dot{g}_m[v_m] w_{m2}[0] \right) - \left( \sum_{m=1}^2 b_{1m} \dot{g}_m[v_m] w_{m2}[0] \sum_{m=1}^2 b_{2m} \dot{g}_m[v_m] w_{m1}[0] \right)$$

from which, it is possible to obtain the final expression of the Jacobian of each block in  $\mathbb{R}^2$

$$\det(J^{\text{small}}) = \dot{g}_1[v_1] \dot{g}_2[v_2] \det(\mathbf{B}) \det(\mathbf{W}[0]). \quad (59)$$

For the particular block structure of the global Jacobian matrix:

$$\det(J) = \prod_{i=1}^M \det(J^{\text{small}}) = \prod_{i=1}^M \dot{g}_1[v_1[i]] \dot{g}_2[v_2[i]] \det(\mathbf{W}[0]) \det(\mathbf{B}). \quad (60)$$

Eq. (60) is the closed form of Jacobian in  $\mathbb{R}^2$ . Eqs. (59) and (60) can be extended to  $\mathbb{R}^N$ .

The extension of (59) is

$$\det(J^{\text{small}}) = \prod_j^N \dot{g}_j[v_j[i]] \det(\mathbf{W}[0]) \det(\mathbf{B}). \quad (61)$$

It is possible to obtain (61) considering that:

- each term which does not contain the product of all the  $N$  terms  $g_h[v_h]$  like  $\dot{g}_1[v_1[i]] \cdot \dot{g}_2[v_2[i]] \cdot \dots \cdot \dot{g}_N[v_N[i]]$  but contains  $\dot{g}_1[v_1[i]]^2 \cdot \dot{g}_3[v_3[i]] \cdot \dots \cdot \dot{g}_N[v_N[i]]$  can be simplified;
- terms which contain  $\dot{g}_1[v_1[i]] \cdot \dot{g}_2[v_2[i]] \cdot \dots \cdot \dot{g}_N[v_N[i]]$  can be grouped together;
- among the terms obtained by grouping  $\dot{g}_1[v_1[i]] \cdot \dot{g}_2[v_2[i]] \cdot \dots \cdot \dot{g}_N[v_N[i]]$ , it is possible to isolate the products of the determinant of matrix  $\mathbf{B}$  and of matrix  $\mathbf{W}[0]$ .

From the above consideration it is possible to derive the following:

$$\det(J) = \prod_{i=1}^M \det(J^{\text{small}}) = \prod_{i=1}^M \left\{ \prod_j^N \dot{g}_j[v_j[i]] \det(\mathbf{W}[0]) \det(\mathbf{B}) \right\}. \quad (62)$$

Eq. (62) is the expression of the Jacobian used in (21).  $\square$

## References

- [1] C. Jutten, J. Herald, Blind separation of sources, Part I: an adaptive algorithm based on neurometric architecture, *Signal Processing* 24 (July 1991) 1–10.
- [2] P. Comon, Independent component analysis, A new concept? *Signal Processing* 36 (1994) 287–314.
- [3] D. Shobben, P. Sommen, A new blind signal separation algorithm based on second order statistics, in: *Proceedings of the IASTED: International Conference on Signal and Image Processing, Las Vegas, USA, October 27–31, 1998*.
- [4] A. Bell, T. Sejnowski, An Information-maximisation approach to blind separation and blind deconvolution, *Neural Comput.* 7 (5) (1995) 1129–1159.
- [5] S. Amari, S. Douglas, A. Cichocki, H. Yang, Multichannel blind deconvolution and equalization using the natural gradient, in: *Proceedings of the First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless, Paris, April 1997*, pp. 101–104.
- [6] J.-F. Cardoso, Infomax and maximum likelihood for blind source separation, *IEEE Lett. Signal Processing* 4(4) (April 1997) 112–114.

- [7] D.T. Pham, D. Garrat, C. Jutter, Separation of mixture of mixture of independent sources through maximum likelihood approach, in: Proceedings of the EUSIPCO, Brussel, Belgium, August 1992, pp. 771–774.
- [8] H. Hyvarinen, Fast and robust fixed-point algorithm for independent component analysis, *Trans. Neural Networks* 10(3) (May 1999) 626–634.
- [9] J.-F. Cardoso, B.I. Laheld, Equivariant adaptive source separation, *IEEE Trans. Signal Process.* 44 (12) (December 1996) 3017–3030.
- [10] T.W. Lee, M. Girolami, A.J. Bell, T.J. Sejnowsky, A unifying information-theoretic framework for independent component analysis, *Int. J. Comput. Math. Appl. Comput.* (1999).
- [11] S. Choi, A. Cichocki, S. Amari, Flexible independent component analysis, *J. VLSI Signal Process. Systems Image Video Technol.* X (May 2000) 1.
- [12] S. Fiori, Blind signal processing by the adaptive activation function neuron, *Neural Networks* 13 (6) (August 2000) 597–611.
- [13] N. Charkani, Y. Deville, Optimization of the asymptotic performance of time domain convolutive sources separation algorithm, in: Proceedings of the ESANN, Bruges, Belgium, April 1997, pp. 273–278.
- [14] T.W. Lee, B. Koehler, A. Orglmeister, Blind source separation of nonlinear mixing models, *Neural Network Signal Process.* VII (1997).
- [15] P. Pajunen, A. Hyvarinen, Nonlinear blind source separation by self-organizing maps, *Proceedings in ICONIP, Hong Kong*, vol. 2, September 1996, pp. 1207–1210.
- [16] A. Hyvarinen, P. Pajunen, Nonlinear independent component analysis: existence and uniqueness results, *Neural Networks* 12 (2) (1999) 429–439.
- [17] Y. Tan, J. Wang, J.M. Zurada, Nonlinear blind source separation using radial basis function, *IEEE Trans. Neural Networks* 12 (1) (January 2001) 124–134.
- [18] A. Taleb, A generic framework for blind source separation in structured nonlinear models, *IEEE Trans. Signal Process.* 50 (8) (August 2002).
- [19] A. Taleb, C. Jutten, Source separation in post nonlinear mixtures, *Trans. Signal Process.* 47 (10) (August 1999).
- [20] M. Solazzi, R. Parisi, A. Uncini, Blind source separation in nonlinear mixtures by adaptive spline neural network, in: Proceedings of the Third Workshop on Independent Component Analysis and Signal Separation (ICA2001), San Diego, CA, USA, 2001.
- [21] M.B. Zade, C. Jutten, K. Najeb, Blind separating, convolutive post nonlinear mixture, ICA 2001, in: Proceedings of the Third Workshop on Independent Component Analysis and Signal Separation (ICA2001), San Diego, CA, USA, 2001, pp. 138–143.
- [22] F. Milani, M. Solazzi, A. Uncini, Blind source separation of convolutive nonlinear mixtures by flexible spline nonlinear functions, *Proceedings of the IEEE International Conference on Acoustic Speech and Signal Proceedings of the ICASSP'02, Orlando, USA*, May 2002.
- [23] M. Solazzi, F. Piazza, A. Uncini, Nonlinear blind source separation by spline neural network, in: ICASSP 2001, Salt Lake city, USA, May 8–11, 2001.
- [24] C. Jutten, J. Karhunen, Advances in nonlinear blind sources separation, in: Proceedings of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan, April 2003.
- [25] D. Vigliano, A. Uncini, Flexible ICA solution for a novel nonlinear blind source separation problem, *IEE Electron. Lett.* 39 (22) (30 October 2003) 1616–1617.
- [26] R.H. Lambert, Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures. A dissertation presented to the Faculty of the Graduate School, University of Southern California, Department of Electrical Engineering, May 1996.
- [27] D. Vigliano, A. Uncini, “Mirror Model” gives separation of convolutive mixing of PNL mixtures, *IEE Electron. Lett.* 40 (7) (April 2004).
- [28] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill International Edition, third edition, 1991.
- [29] S. Amari, S.C. Douglas, Why natural gradient, in: Proceedings of the IEEE ICASSP, vol. II, Seattle, WA, May 1998, p. 1216.
- [30] S. Choi, A. Cichocki, S. Amari, Two spatio-temporal decorrelation learning algorithms and their application to multichannel blind deconvolution, in: Proceedings of the ICASSP, Phoenix, AZ, March 1999, pp. 1085–1088.
- [31] S. Amari, Information geometry on hierarchy of probability distribution, *IEEE Trans. Inform. Theory* 47 (5) (July 2001).
- [32] S. Choi, S. Amari, A. Cicochi, Natural gradient learning for spatio-temporal decorrelation: recurrent network, *IEICE Trans. Fund.* E83-A (12) (December 2000).
- [33] S. Choi, H. Hong, H. Glotin, F. Bertommier, Multichannel signal separation for cocktail party speech recognition: a dynamic recurrent network, in: Proceedings of the ICSLP, Beijing, China, 2000.
- [34] S. Choi, A. Cichocki, Blind signal deconvolution by spatio temporal decorrelation and demixing, in: IEEE Workshop on Neural Networks for Signal Processing, Ameldia Island, FL, September 1997.
- [35] S. Choi, A. Cichocki, Adaptive blind separation of speech signals: cocktail party problem, in: International Conference on Speech Processing (ICSP'97), Seoul, Korea, August 26–28, 1997, pp. 617–622.
- [36] S. Amari, Natural gradient works efficiently in learning, *Neural Comput.* 10 (1998) 251–276.

- [37] A. Uncini, F. Piazza, Blind signal processing by complex domain adaptive spline neural network, *Trans. Neural Networks* 14 (2) (March 2003).
- [38] A. Uncini, L. Vecchi, P. Campolucci, F. Piazza, Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization, *IEEE Trans. Signal Process.* 47 (2) (February 1999).
- [39] S. Guarnieri, F. Piazza, A. Uncini, Multilayer feedforward networks with adaptive spline activation function, *IEEE Trans. Neural Networks* 10 (3) (May 1999) 672–683.
- [40] A. Uncini, L. Vecchi, F. Piazza, Learning and approximation capabilities of adaptive Spline activation function neural network, *Neural Network* 11 (2) (March 1998) 259–270.
- [41] M. Solazzi, F. Piazza, A. Uncini, An adaptive spline nonlinear function for blind signal processing, in: *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, vol. X, December 2000, pp. 396–404.
- [42] M. Cohen, G. Cauwenberghs, Blind separation of linear convolutive mixtures through parallel stochastic optimization, in: *Proceedings of the ISCAS*, 1998.
- [43] T.W. Lee, A. Bell, R.H. Lambert, Blind separation of delayed and convolved sources, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), *Advances in Neural Information Processing Systems*, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 758–764.
- [44] T.W. Lee, A. Bell, R. Orglmeister, Blind source separation of real world signals, in: *Proceedings of the International Conference on Neural Networks (ICNN)*, Houston, June 1997.
- [45] I. Sabala, A. Cichocki, Relationships between instantaneous blind source separation and multichannel blind deconvolution, in: *Proceedings of the IJCNN*, 1998.
- [46] K. Torkkola, Blind deconvolution, information maximization, and recursive filters, in: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Munich, Germany, April 21–24, 1997, pp. 3301–3304.
- [47] K. Torkkola, Blind separation of audio signals: Are we there yet? in: *Proceedings of the ICA and BSS*, Aussois, France, January 11–15, 1999, pp. 239–244.
- [48] K. Torkkola, Blind separation of convolved sources based on information maximization, in: *IEEE Workshop on Neural Networks for Signal Processing*, Kyoto, Japan, September 4–6, 1996, pp. 423–432.
- [49] D. Shobben, K. Torkkola, P. Smaragdis, Evaluation of blind signal separation methods, in: *Proceedings of the ICA and BSS*, Aussois, France, January 11–15, 1999.
- [50] S.M. Kay, S.L. Marple, Spectrum analysis: a modern perspective, *Proc. IEEE* 69 (11) (1981) 1380–1418.
- [51] C.G.A Prieto, J. Ortega, New geometrical approach for blind separation of sources mapped to a neural network, in: *Proceedings of the NICROSP-96, International Workshop on Neural Networks for Identification, Control, Robotics, and Signal-Image Processing*, 21–23 August, 1996, Venecia, Italia.