Authors' final version of a paper

published in "Signal Processing"

Paper reference:

S. Hosseini, Y. Deville, "Recurrent networks for separating extractable-target nonlinear mixtures. Part II: Blind configurations", Signal Processing, vol. 93, Issue 4, pp. 671-683, April 2013.

Elsevier on-line version:

http://dx.doi.org/10.1016/j.sigpro.2012.08.027

Recurrent networks for separating extractable-target nonlinear mixtures. Part II: Blind configurations

Shahram Hosseini*, Yannick Deville

Institut de Recherche en Astrophysique et Planétologie, Université de Toulouse, UPS-CNRS-OMP, 14 avenue Edouard Belin, 31400 Toulouse, France.

Abstract

While most reported blind source separation methods concern linear mixtures, we here address the nonlinear case. In the first part of this paper, we introduced a general class of nonlinear mixtures which can be inverted using recurrent networks. That part was focused on separating structures themselves and therefore on the non-blind configuration, whereas the current paper addresses the estimation of the parameters of this large class of structures in a blind context. We propose a maximum likelihood approach to this end. The main advantage of this approach is that it exploits the knowledge of the parametric model of mixing transformation in the separation procedure while its implementation does not require the knowledge of the *explicit inverse* of the model because the separating structure can be designed using a recurrent network. In particular, we illustrate in detail the proposed approach for a linear-quadratic mixture by using an extended recurrent network with self-feedback parameters which guarantee its local stability. Simulation results show the very good performance of the proposed algorithm.

Keywords: Blind source separation, Independent component analysis, Nonlinear mixture, Linear-quadratic mixture, Maximum likelihood

*Corresponding author, Tel.: +33-5-61-33-28-79; Fax: +33-5-61-33-28-40.
Email addresses: Shahram.Hosseini@irap.omp.eu (Shahram Hosseini), Yannick.Deville@irap.omp.eu (Yannick Deville)

1. Introduction

Blind Source Separation (BSS) aims at restoring source signals from their mixtures when the mixing parameters are unknown. The main class of methods proposed to this end is based on the assumed statistical independence of the sources and is called Independent Component Analysis (ICA). While linear BSS has been extensively studied (see for example [1] and references therein), much less work is available on nonlinear BSS. Several authors have considered general nonlinear mixtures (see e.g. [2, 3, 4, 5, 6]). These works are not dedicated to a specific class of mixtures and do not take advantage of possible knowledge of the parametric model of mixing transformations. For example, the MISEP approach proposed in [4] tries to invert the mixing model using a multilayer perceptron. The main problem of these approaches concerns the non-uniqueness of nonlinear ICA: it is well known that the independence hypothesis is not sufficient for separating general nonlinear mixtures because of the very large indeterminacies which make the problem ill-posed [7, 8]. A natural idea for reducing these indeterminacies is to constrain the structure of mixing and separating models to belong to a certain set of transformations. This supplementary constraint can be viewed as a regularization of the initially ill-posed problem [9, 10]. Thus, several authors have studied specific classes of nonlinear mixing models like post-nonlinear mixtures [8, 11, 12], linear-quadratic mixtures [13, 14, 15, 16, 17, 18, 19], linearizable mappings [20] or nonlinear mixtures encountered on gas or chemical sensors [21, 22] (see also [9] for a more general framework). More references about nonlinear BSS can be found in Chapter 14 of [1].

Consider the mixing equation

$$\mathbf{x}(n) = \boldsymbol{\mathcal{F}}(\mathbf{s}(n), \boldsymbol{\theta}) \tag{1}$$

where $\mathbf{s}(n) = [s_1(n), \cdots, s_N(n)]^T$ is the vector of N independent unknown sources at time n (^{T} denotes transposition), $\mathbf{x}(n) = [x_1(n), \cdots, x_P(n)]^T$ is the vector of P observations and $\boldsymbol{\mathcal{F}}$ is a memoryless parametric function, defined by the unknown parameter vector $\boldsymbol{\theta}$. Even when the mixing model and its

parameter values are known (i.e. in the non-blind configuration) it is not always straightforward to create a system which implements the inverse of this model. In the first part of this article [23], we introduced a general class of nonlinear mixtures called "extractable-target mixtures" (ETM) which can be inverted using recurrent networks. In its general form, an ETM is described by the following mixing equations

$$x_i(n) = f(T_i[\mathbf{s}(n)], I_i[\mathbf{s}(n)]) \quad \forall i \in \{1, \cdots, P\} \tag{2}$$

where $T_i[\mathbf{s}(n)]$ and $I_i[\mathbf{s}(n)]$ are, respectively, the target and interfering terms of the considered model, i.e. the components of $x_i(n)$ that we aim at keeping and removing in the network outputs and f is a quite general function (see [23] for more details). A particular case of these mixtures, called "additive-target mixtures" (ATM) in [23], is described by

$$x_i(n) = T_i[\mathbf{s}(n)] - I_i[\mathbf{s}(n)] \quad \forall i \in \{1, \cdots, P\}. \tag{3}$$

In a simpler configuration, the target terms $T_i[\mathbf{s}(n)]$ are equal to the source signals $s_i(n)$ up to classical scale indeterminacy: $T_i[\mathbf{s}(n)] = s'_i(n) = k_i s_i(n)$ where k_i is a constant. In this case, the mixture reads

$$x_i(n) = s'_i(n) - I_i[\mathbf{s}(n)] \quad \forall i \in \{1, \cdots, P\}. \tag{4}$$

In [23], we explained how recurrent networks may be used to retrieve the sources from their ETM in a non-blind configuration. In particular, for the mixing model (4) with $N = P = 2$, we proposed a recurrent network (see Fig. 1) and showed that by choosing the "cancellation operators" C_i such that

$$C_i[\mathbf{s}'(n)] = I_i[\mathbf{s}(n)], \tag{5}$$

the normalized sources $\mathbf{s}'(n)$ correspond to a fixed point of the recurrent network. We also proposed an extended structure (see Fig. 2) by adding self-feedback operators F_i from each output to each input having the same index i. This extension provides additional flexibility which may be used to improve the stability of the recurrent network. We especially showed in [23] how the

stability of the extended network at its main fixed point can be guaranteed when the observations are linear-quadratic mixtures of the sources with known mixing parameters.

While Part I of this paper was focused on separating structures themselves and therefore on the non-blind configuration, the current paper addresses the estimation of the parameters of such structures in a blind context. Our estimation framework is the maximum likelihood (ML) approach. This approach for linear BSS was initially proposed in [24] and was developed and completed by Pham and Garat [25]. Cardoso [26] showed the equivalence between the ML approach and the Infomax [27] algorithm. It is well known that the ML approach is closely related to the Mutual Information approach which has been used for separating linear [28, 29] and nonlinear [9, 4, 30] mixtures.

In a conference paper [16], we proposed an ML approach for blindly separating linear-quadratic mixtures using a basic recurrent network. The main drawback of that proposed approach was the possible instability of the network: during the learning procedure, the network parameters could take values making the network unstable. In the current paper, the approach proposed in [16] is extended from two points of view. First, we propose a general ML approach for blindly separating any ETM mixtures. A main advantage of our proposed approach results from an original use of the implicit differentiation, which allows us to derive the analytical expression of the gradient without requiring the knowledge of the explicit inverse of the mixing model. Second, we propose a method to avoid the problems related to the use of the recurrent network and, in particular, to guarantee its stability during the learning procedure when using it for blindly separating 2×2 linear-quadratic mixtures. We also provide new simulation results which validate the efficiency of this new proposed method.

The remainder of this paper is organized as follows. In Section 2, we explain how to blindly separate the sources mixed by an ETM using a maximum likelihood approach. In Section 3, we illustrate the proposed procedure using a linear-quadratic mixture. Experimental results are reported in Section 4 and Section 5 concludes this work.

2. Blind separation of general ETM

Consider the mixing equation (1). In the following, we suppose that:

1. The mixture is determined, i.e. $P = N$.
2. Each source $s_i(n)$ is an independent and identically distributed (i.i.d.) random signal[1] with probability density function (pdf) $f_{S_i}(.)$. As a result, each observed signal $x_i(n)$ is i.i.d. too.
3. The sources are statistically independent so that $f_{\mathbf{S}}(.) = \prod_{i=1}^{N} f_{S_i}(.)$.

The maximum likelihood principle can be applied as follows. We denote by $\hat{\boldsymbol{\Theta}}$ the set of all parameter column vectors $\hat{\boldsymbol{\theta}}$ such that the model (1) is bijective in the variation domain of the sources. The inverse of this model for each vector $\hat{\boldsymbol{\theta}}$ will be denoted by $\hat{\mathbf{s}}(n) = \boldsymbol{\mathcal{F}}^{-1}(\mathbf{x}(n), \hat{\boldsymbol{\theta}})$.

Once the source pdf $f_{\mathbf{S}}(.)$ has been fixed, the distribution of the transformed vector $\boldsymbol{\mathcal{F}}(\mathbf{s}, \hat{\boldsymbol{\theta}})$ only depends on $\hat{\boldsymbol{\theta}}$. We use this family of distributions as a parametric model for the pdf of observations[2] and denote it by $\mathcal{P} = \{p_{\hat{\boldsymbol{\theta}}}\ ,\ \hat{\boldsymbol{\theta}} \in \hat{\boldsymbol{\Theta}}\}$.

Given K samples of the observed signals $\boldsymbol{x}(n)$, the likelihood that these samples are drawn with a particular pdf $p_{\hat{\boldsymbol{\theta}}}$ is given by

$$L = \prod_{n=1}^{K} p_{\hat{\boldsymbol{\theta}}}(x_1(n), \cdots, x_N(n)), \tag{6}$$

because the signals are supposed to be i.i.d.[3]

The mixture being supposed to be bijective, we can write

$$L = \prod_{n=1}^{K} \frac{f_{\mathbf{S}}(\hat{s}_1(n), \cdots, \hat{s}_N(n))}{|\hat{J}|} = \prod_{n=1}^{K} \frac{\prod_{i=1}^{N} f_{S_i}(\hat{s}_i(n))}{|\hat{J}|} \tag{7}$$

[1] As mentioned in [25], this is only a working hypothesis to simplify the likelihood function. This does not mean that the proposed method cannot be used for separating non-i.i.d. signals but means we chose not to exploit the time structure of signals in the separating procedure.

[2] It is clear that this parametric pdf will be equal to the actual observation pdf if $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

[3] The above formulation of the ML estimation was used by Cardoso in the case of linear mixtures [26].

where $\hat{J}$ is the Jacobian of the mixing transformation (1), i.e. $det\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \mathbf{s}}$, evaluated at $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{s}}(n) = \boldsymbol{\mathcal{F}}^{-1}(\mathbf{x}(n), \hat{\boldsymbol{\theta}})$. The maximum likelihood estimation of the actual parameters $\boldsymbol{\theta}$ consists in maximizing this likelihood with respect to $\hat{\boldsymbol{\theta}}$. This is equivalent to the minimization of $C = \frac{-1}{K}\log L = \frac{-1}{K}\sum_{n=1}^{K}\log\frac{\prod_{i=1}^{N} f_{S_i}(\hat{s}_i(n))}{|\hat{J}|}$ which can be denoted using the time average operator $E_K[.]$ as

$$C = -E_K[\sum_{i=1}^{N}\log f_{S_i}(\hat{s}_i)] + E_K[\log|\hat{J}|]. \tag{8}$$

Minimizing this cost function requires that its gradient with respect to the parameter vector $\hat{\boldsymbol{\theta}}$ vanishes. Defining the score function of the source s_i by

$$\psi_{S_i}(s_i) = -d\log f_{S_i}(s_i)/ds_i, \tag{9}$$

and considering that $\frac{\partial \log|\hat{J}|}{\partial\hat{\boldsymbol{\theta}}} = \frac{1}{\hat{J}}\frac{\partial\hat{J}}{\partial\hat{\boldsymbol{\theta}}}$, the gradient reads

$$\frac{dC}{d\hat{\boldsymbol{\theta}}} = \Big(\sum_{i=1}^{N} E_K\left[\psi_{S_i}(\hat{s}_i)\frac{\partial\hat{s}_i}{\partial\hat{\boldsymbol{\theta}}}\right]\Big) + E_K\left[\frac{1}{\hat{J}}\frac{\partial\hat{J}}{\partial\hat{\boldsymbol{\theta}}}\right]. \tag{10}$$

According to $\mathbf{x} = \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}, \hat{\boldsymbol{\theta}})$ and considering $\hat{\boldsymbol{\theta}}$ as an independent variable and $\hat{\mathbf{s}}$ as a dependent variable, we can write using implicit differentiation

$$\mathbf{0} = \frac{\partial\hat{\mathbf{s}}}{\partial\hat{\boldsymbol{\theta}}}\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\mathbf{s}}} + \frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}\ cst} \tag{11}$$

which yields

$$\frac{\partial\hat{\mathbf{s}}}{\partial\hat{\boldsymbol{\theta}}} = -\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}\ cst}(\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\mathbf{s}}})^{-1}. \tag{12}$$

In the above expressions, "$\hat{\mathbf{s}}$ *cst*" stands for "$\hat{\mathbf{s}}$ is supposed to be constant". Note that $\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\mathbf{s}}}$ is the $N\times N$ Jacobian matrix of the mixing model, with the generic entry $\left(\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\mathbf{s}}}\right)_{i,j} = \frac{\partial\mathcal{F}_j}{\partial\hat{s}_i}$ and the entry (i,j) of the matrix $\frac{\partial\boldsymbol{\mathcal{F}}}{\partial\hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}\ cst}$ (respectively $\frac{\partial\hat{\mathbf{s}}}{\partial\hat{\boldsymbol{\theta}}}$) is $\frac{\partial\mathcal{F}_j}{\partial\hat{\theta}_i}_{|\hat{\mathbf{s}}\ cst}$ (respectively $\frac{\partial\hat{s}_j}{\partial\hat{\theta}_i}$). Inserting the components of (12) in (10), we obtain the expression of the gradient which may be used for minimizing the cost function using e.g. a gradient descent algorithm

$$\hat{\boldsymbol{\theta}}_{new} = \hat{\boldsymbol{\theta}}_{old} - \mu\frac{dC}{d\hat{\boldsymbol{\theta}}} \tag{13}$$

where μ is a positive learning rate. The main advantage of the above approach based on the implicit differentiation is that it does not require the knowledge

of the explicit inverse of the mixing model. In fact, according to (10) and (12), in order to obtain the *analytical expression* of the gradient $\frac{dC}{d\hat{\boldsymbol{\theta}}}$, we only need to know the expression of the mixing function $\boldsymbol{\mathcal{F}}$ and not its inverse.

However, to compute the *numerical value* of the gradient from the expression (10), we need the signal samples $\hat{\mathbf{s}}(n) = \boldsymbol{\mathcal{F}}^{-1}(\mathbf{x}(n), \hat{\boldsymbol{\theta}})$ for the current value of $\hat{\boldsymbol{\theta}}$ at each iteration of the gradient descent algorithm. As mentioned in the previous section, it is not always straightforward to find the inverse of a mixing function, but in the case of an ETM, it is possible to design such a "separating" structure using a recurrent network without the knowledge of the explicit inverse of the mixing model.

In practice, the pdf $f_{S_i}(.)$ and consequently the score functions $\psi_{S_i}(.)$ of the actual sources are usually unknown. Like in linear BSS [25], we can replace them by the estimated score functions of the signals $\hat{s}_i(n)$ (obtained at the outputs of the separating structure) in each iteration of the gradient algorithm. Using this estimation of the score functions, maximizing the likelihood is equivalent to minimizing the mutual information of the output signals $\hat{s}_i(n)$, in the same manner as in linear BSS [29].

These score functions may be for example estimated using the approach proposed in [25] which consists in writing $\psi_{S_i}(\hat{s}_i) = \sum_{m=1}^{M} c_{im}\phi_m(\hat{s}_i)$, where $\phi_m(\hat{s}_i)$ are some basis functions, and in computing the coefficients c_{im} by solving the following equation

$$\mathbf{G}_i[c_{i1}, \cdots, c_{iM}]^T = \mathbf{g}_i \tag{14}$$

where $\mathbf{G}_i = E[\boldsymbol{\phi}(\hat{s}_i)\boldsymbol{\phi}(\hat{s}_i)^T]$, $\mathbf{g}_i = E[\boldsymbol{\phi}'(\hat{s}_i)]$ with $\boldsymbol{\phi}(\hat{s}_i) = [\phi_1(\hat{s}_i), \cdots, \phi_M(\hat{s}_i)]^T$ and $\boldsymbol{\phi}'(\hat{s}_i)$ its derivative with respect to $\hat{s}_i$.

At each step of the gradient algorithm, once a new estimate of the mixing parameters has been computed, it may be used for determining the parameters of a recurrent network so that its outputs provide a new estimate of the sources $\hat{s}_i$ (see Fig. 1 and 2). Therefore, we can use the following algorithm for blindly separating an ETM.

- Initialize the estimated mixing parameters $\hat{\boldsymbol{\theta}}$ (*e.g.* with random values if there is no information about them).

repeat

- Initialize the parameters of the separating recurrent network using the current values of the estimated mixing parameters.
- Iterate this network until convergence to obtain a new estimate of the sources.
- Estimate the score functions of the estimated sources.
- Compute the gradient.
- Update the estimated mixing parameters.

until the convergence of the estimated mixing parameters

In the following section, we illustrate this proposed algorithm for a linear-quadratic mixture.

3. Blind separation of linear-quadratic mixtures

In this section, we study a linear-quadratic mixing model which may be considered as the simplest (nonlinear) version of a general polynomial model. This model has recently been used to describe the show-through effect in scanned documents [17, 31, 19]. When both sides of a thin paper are printed, each side is a mixture of the front and back images due to transparency. Thus, we have two mixtures of two sources. This problem was first addressed by Almeida [32] who used a general-purpose neural network for separating the sources. Almeida and Almeida [17, 19] and Merrikh-Bayat et al. [31] showed that the mixing effect in the show-through problem may be approximated using more general linear-quadratic models (a bi-affine model in [17, 19] and a linear-quadratic model with filtering blocks to take into account the blurring effects in [31]).

The linear-quadratic model is also encountered in hyperspectral remote sensing [33, 34, 35, 36]. When the surfaces are not flat and homogeneous, the 3D structure induces multiple scattering of light between surfaces, which yields a linear-quadratic model when only taking into account second-order interactions.

In a more general context, the linear-quadratic model may be considered as a truncated Taylor series which may be used to approximate unknown nonlinear mixtures. In the remainder of this section, we only study the case of $N = P = 2$ (i.e. two mixtures of two sources). This case was investigated in the first part of this article ([23]) as an illustrative example to show how the stability problem can be treated in ETMs. In the current paper, we aim at showing how the results of [23] can be adapted to the blind configuration.
Suppose u_1 and u_2 are two independent random signals[4]. Given the following nonlinear instantaneous mixing model

$$x_i = \rho_{i1}u_1 + \rho_{i2}u_2 + \xi_i u_1 u_2 \qquad i = 1, 2 \tag{15}$$

we would like to estimate u_1 and u_2 up to a permutation and a scale factor (and possibly an additive constant). For simplicity, let us denote $s_1 = \rho_{11}u_1$ and $s_2 = \rho_{22}u_2$. The signals s_1 and s_2 will be referred to as the *sources* in the following. Thus, (15) can be rewritten as

$$\begin{aligned} x_1 &= s_1 - L_{12}s_2 - Q_1 s_1 s_2 \\ x_2 &= s_2 - L_{21}s_1 - Q_2 s_1 s_2 \end{aligned} \tag{16}$$

in which $L_{12} = -\rho_{12}/\rho_{22}$ and $L_{21} = -\rho_{21}/\rho_{11}$ represent the linear contributions of the sources in the mixture, and $Q_1 = -\xi_1/(\rho_{11}\rho_{22})$ and $Q_2 = -\xi_2/(\rho_{11}\rho_{22})$ represent the quadratic contributions. The negative signs are chosen for simplifying the notations of the separating structure.

A more general form of the model (15), containing the additional terms ${u_1}^2$ and ${u_2}^2$, has been studied by a few authors [13], [14], for the special case of *circular* complex sources, when at least 5 mixtures are available. Another linear-quadratic model, similar to (15), has also been used in [18] with two *binary* sources and at least 3 mixtures. In the current work, however, we suppose that:

[4]In [23], we used the notation s_i for the original sources and s'_i for the normalized ones. In the following, however, the original sources will be denoted by u_i and the normalized sources by s_i. This choice improves the readability of equations.

1) the sources can be any arbitrary real signals, and 2) only two mixtures are available.

3.1. Comments about the invertibility of the model

As shown in [15], multiplying the first equation of (16) by $-Q_2$ and the second equation by Q_1, then adding the results, yields

$$s_2 = [Q_1 x_2 - Q_2 x_1 + (Q_1 L_{21} + Q_2) s_1]/(Q_2 L_{12} + Q_1).$$

Substituting this expression of s_2 in the second equation of (16), leads to the following equation of the second degree for s_1

$$(Q_1 L_{21} + Q_2) s_1^2 + (Q_1 x_2 - Q_2 x_1 + L_{12} L_{21} - 1) s_1 + (x_1 + L_{12} x_2) = 0.$$

Similarly, the corresponding equation with respect to s_2 is

$$(Q_2 L_{12} + Q_1) s_2^2 + (Q_2 x_1 - Q_1 x_2 + L_{12} L_{21} - 1) s_2 + (x_2 + L_{21} x_1) = 0.$$

Solving the above equations for s_1 and s_2 leads to the following two pairs of solutions:

$$\begin{aligned} (\int_1, \int_2)_1 &= ((-b_1 + \sqrt{\Delta_1})/2a_1, (-b_2 + \sqrt{\Delta_2})/2a_2) \\ (\int_1, \int_2)_2 &= ((-b_1 - \sqrt{\Delta_1})/2a_1, (-b_2 - \sqrt{\Delta_2})/2a_2) \end{aligned} \tag{17}$$

where $\Delta_i = b_i^2 - 4a_i c_i$, $a_1 = Q_2 + L_{21} Q_1$, $a_2 = Q_1 + L_{12} Q_2$, $b_1 = Q_1 x_2 - Q_2 x_1 + L_{12} L_{21} - 1$, $b_2 = Q_2 x_1 - Q_1 x_2 + L_{12} L_{21} - 1$, $c_1 = x_1 + L_{12} x_2$ and $c_2 = x_2 + L_{21} x_1$. It can be easily verified that $\Delta_1 = \Delta_2 = J^2$, where J is the Jacobian of the mixing model (16) and reads

$$J = 1 - L_{12} L_{21} - (Q_2 + L_{21} Q_1) s_1 - (Q_1 + L_{12} Q_2) s_2. \tag{18}$$

It is clear that the linear-quadratic model is not bijective in $\mathbb{R}^2$. However, if the variation domain of the sources is limited, the mixture may be bijective in this variation domain, which will be denoted by Ω in the following. In fact, according to the variation domain of the two sources, three different cases may be considered:

1) $J < 0$ everywhere in Ω. In this case (17) becomes:

$$(\int_1, \int_2)_1 = (s_1, s_2) \tag{19}$$

$$(\int_1, \int_2)_2 = (-\frac{Q_1 + L_{12}Q_2}{Q_2 + L_{21}Q_1}s_2 - \frac{L_{12}L_{21} - 1}{Q_2 + L_{21}Q_1}, -\frac{Q_2 + L_{21}Q_1}{Q_1 + L_{12}Q_2}s_1 - \frac{L_{12}L_{21} - 1}{Q_1 + L_{12}Q_2}). \tag{20}$$

Thus, the first pair of solutions in (17) corresponds to the actual sources and the second one to another solution, equivalent to the first one up to a permutation, a scale factor, and an additive constant.

We now show that the mixture is bijective on Ω in this first case. Consider the function

$$J_1((\int_1, \int_2)) = 1 - L_{12}L_{21} - (Q_2 + L_{21}Q_1)\int_1 - (Q_1 + L_{12}Q_2)\int_2. \tag{21}$$

Replacing $(\int_1, \int_2)$ first by $(\int_1, \int_2)_1$, then by $(\int_1, \int_2)_2$ defined above, and comparing the results with (18), it can be easily verified that

$$J_1((\int_1, \int_2)_1) = -J_1((\int_1, \int_2)_2) = J \tag{22}$$

The line $J_1((\int_1, \int_2)) = 0$ splits the $(\int_1, \int_2)$ plane in two parts. If $J < 0$ for all the source values (i.e. in Ω), then $(\int_1, \int_2)_1 = (s_1, s_2)$ is always situated in the one part of the $(\int_1, \int_2)$ plane corresponding to $J < 0$ (thus containing Ω) while $(\int_1, \int_2)_2$ is situated in the other part corresponding to $J > 0$ so that the model is bijective on Ω. [5]

2) $J > 0$ everywhere in Ω. In this case, it may be shown that the first pair of solutions in (17) corresponds to the permuted sources, defined by the right-hand side of (20), and the second one corresponds to the actual sources (s_1, s_2). An example is shown in Fig. 3 for the numerical values $L_{12} = -0.2$, $L_{21} = 0.2$, $Q_1 = -0.8$, $Q_2 = 0.8$ and $s_i \in [-0.5, 0.5]$.

[5] Note that contrary to the linear-quadratic mixture, a general mixing model may be non-bijective on a region, even if the sign of its Jacobian does not change on that region. For example [37], the mapping defined by $x_1 = e^{2s_1} - s_2^2 + 3$, $x_2 = 4e^{2s_1}s_2 - s_2^3$ has a positive Jacobian $J = 2e^{2s_1}(4e^{2s_1} + 5s_2^2)$ in $\mathbb{R}^2$ while both points $(0, 2)$ and $(0, -2)$ are mapped to the origin.

3) $J > 0$ for some values in Ω and $J < 0$ for other values. In this case, each solution in (17) corresponds to the non-permuted sources (19) for some values of the observations and to the permuted sources (20) for the other values. Moreover, the mixture is not bijective in the variation domain of the sources. An example is shown in Fig. 4 (with the same coefficients as in the second case, but for $s_i \in [-2, 2]$). The permutation effect is visible in the figure. Thus, it is clear that even in the non-blind configuration, the actual source signals may be retrieved only if the Jacobian of the mixing model is always negative or always positive, *i.e.* for all the source values. Otherwise, although the sources are separated *sample by sample*, each retrieved signal contains successive samples of the two sources. This theoretically insoluble problem which arises because the mixing model (16) is not bijective in $\mathbb{R}^2$ should not discourage us. In fact, in real-world applications, the mixture may be bijective in the variation domain of the sources[6]. Moreover, in a more elaborate scheme, the structure of the sources (auto-correlation, non-stationarity, etc) may be used to solve the permutation problem as in frequency-domain BSS methods applied to convolutive mixtures. Thus, in the following, we suppose that the sources and the mixture coefficients have numerical values ensuring that the Jacobian J of the mixing model has a constant sign.

3.2. Estimation of the mixing parameters

Considering the mixing model (16), we aim at estimating the parameter vector $\boldsymbol{\theta} = [L_{12}, L_{21}, Q_1, Q_2]^T$ using the maximum likelihood approach described in Section 2. Consider the set of parameter vectors $\hat{\boldsymbol{\theta}} = [\hat{L}_{12}, \hat{L}_{21}, \hat{Q}_1, \hat{Q}_2]^T$ for which the mixing model $\mathbf{x}(n) = \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})$ is bijective in the variation domain of the sources, and suppose $\hat{\mathbf{s}}(n) = [\hat{s}_1(n), \hat{s}_2(n)]^T$ is the solution of the following

[6] To better understand this phenomenon, consider for example the simple one-dimensional model $x(n) = s(n)^2$. It is clear that this model is not bijective in general: if the signals contain 1000 samples, there are 2^{1000} different source signals which could produce the observed signal. However, if we know that the source samples are positive, we can retrieve the source signal in a unique manner.

equations in this variation domain for the observed samples:

$$x_1(n) = \hat{s}_1(n) - \hat{L}_{12}\hat{s}_2(n) - \hat{Q}_1\hat{s}_1(n)\hat{s}_2(n),$$
$$x_2(n) = \hat{s}_2(n) - \hat{L}_{21}\hat{s}_1(n) - \hat{Q}_2\hat{s}_1(n)\hat{s}_2(n). \tag{23}$$

Based on (18), the Jacobian of the above model reads

$$\hat{J}(n) = 1 - \hat{L}_{12}\hat{L}_{21} - (\hat{Q}_2 + \hat{L}_{21}\hat{Q}_1)\hat{s}_1(n) - (\hat{Q}_1 + \hat{L}_{12}\hat{Q}_2)\hat{s}_2(n). \tag{24}$$

Using (10) and (12), the gradient of the cost function C in Eq. (8) with respect to the parameter vector $\hat{\boldsymbol{\theta}}$ reads as follows, where the time index n is omitted for simplifying the notation (see Appendix A for the computation details):

$$\frac{dC}{d\hat{\boldsymbol{\theta}}} = E_K\Big[\frac{D_1}{\hat{J}}, \frac{D_2}{\hat{J}}, \frac{D_3}{\hat{J}}, \frac{D_4}{\hat{J}}\Big]^T \tag{25}$$

where

$$\begin{aligned} D_1 &= \Big[\psi_{S_1}(\hat{s}_1)(1 - \hat{Q}_2\hat{s}_1)\hat{s}_2 + \psi_{S_2}(\hat{s}_2)(\hat{L}_{21} + \hat{Q}_2\hat{s}_2)\hat{s}_2\Big] - \Big[\hat{L}_{21} + \hat{Q}_2\hat{s}_2\Big] - \\ &\quad \Big[(\hat{Q}_2 + \hat{L}_{21}\hat{Q}_1)(1 - \hat{Q}_2\hat{s}_1)\hat{s}_2/\hat{J}\Big] - \Big[(\hat{Q}_1 + \hat{L}_{12}\hat{Q}_2)(\hat{L}_{21} + \hat{Q}_2\hat{s}_2)\hat{s}_2/\hat{J}\Big], \end{aligned}$$

$$\begin{aligned} D_2 &= \Big[\psi_{S_1}(\hat{s}_1)(\hat{L}_{12} + \hat{Q}_1\hat{s}_1)\hat{s}_1 + \psi_{S_2}(\hat{s}_2)(1 - \hat{Q}_1\hat{s}_2)\hat{s}_1\Big] - \Big[\hat{L}_{12} + \hat{Q}_1\hat{s}_1\Big] - \\ &\quad \Big[(\hat{Q}_1 + \hat{L}_{12}\hat{Q}_2)(1 - \hat{Q}_1\hat{s}_2)\hat{s}_1/\hat{J}\Big] - \Big[(\hat{Q}_2 + \hat{L}_{21}\hat{Q}_1)(\hat{L}_{12} + \hat{Q}_1\hat{s}_1)\hat{s}_1/\hat{J}\Big], \end{aligned}$$

$$\begin{aligned} D_3 &= \Big[\psi_{S_1}(\hat{s}_1)(1 - \hat{Q}_2\hat{s}_1)\hat{s}_1\hat{s}_2 + \psi_{S_2}(\hat{s}_2)(\hat{L}_{21} + \hat{Q}_2\hat{s}_2)\hat{s}_1\hat{s}_2\Big] - \Big[\hat{L}_{21}\hat{s}_1 + \hat{s}_2\Big] - \\ &\quad \Big[(\hat{Q}_2 + \hat{L}_{21}\hat{Q}_1)(1 - \hat{Q}_2\hat{s}_1)\hat{s}_1\hat{s}_2/\hat{J}\Big] - \Big[(\hat{Q}_1 + \hat{L}_{12}\hat{Q}_2)(\hat{L}_{21} + \hat{Q}_2\hat{s}_2)\hat{s}_1\hat{s}_2/\hat{J}\Big], \end{aligned}$$

$$\begin{aligned} D_4 &= \Big[\psi_{S_1}(\hat{s}_1)(\hat{L}_{12} + \hat{Q}_1\hat{s}_1)\hat{s}_1\hat{s}_2 + \psi_{S_2}(\hat{s}_2)(1 - \hat{Q}_1\hat{s}_2)\hat{s}_1\hat{s}_2\Big] - \Big[\hat{L}_{12}\hat{s}_2 + \hat{s}_1\Big] - \\ &\quad \Big[(\hat{Q}_1 + \hat{L}_{12}\hat{Q}_2)(1 - \hat{Q}_1\hat{s}_2)\hat{s}_1\hat{s}_2/\hat{J}\Big] - \Big[(\hat{Q}_2 + \hat{L}_{21}\hat{Q}_1)(\hat{L}_{12} + \hat{Q}_1\hat{s}_1)\hat{s}_1\hat{s}_2/\hat{J}\Big]. \end{aligned}$$

3.3. Computing the outputs using a recurrent network

Following (25), it is clear that the computation of the numerical value of the gradient at each iteration of the gradient algorithm requires the knowledge of

the signals $\hat{s}_1(n)$ and $\hat{s}_2(n)$ for the current value of $\hat{\boldsymbol{\theta}}$. A natural idea to compute these signals is to form a direct "separating" structure using any of the equations in (17), replacing the actual parameter vector $\boldsymbol{\theta} = [L_{12}, L_{21}, Q_1, Q_2]^T$ by the estimated parameters $\hat{\boldsymbol{\theta}} = [\hat{L}_{12}, \hat{L}_{21}, \hat{Q}_1, \hat{Q}_2]^T$ in these equations. Although this approach may be used for our special mixing model (16), as soon as a more complicated model with unknown explicit inverse is considered, the solutions $(\int_1, \int_2)$ can no longer be analytically determined so that the generalization of the method to arbitrary ETM models becomes impossible. To avoid this limitation, we propose to use recurrent structures as mentioned in the previous sections.

Figure 5 shows the basic and extended recurrent structures that we proposed in [23] for inverting linear-quadratic mixtures. In [23], we analyzed these structures in the non-blind configuration, i.e. when the actual mixing parameter vector $\boldsymbol{\theta}$ is known and the parameter values of the recurrent network are matched to these mixing parameters so that one of the equilibrium points of the network corresponds to the actual sources $\mathbf{s}(n)$ without permutation (but with possible scale factors for the extended network). We showed that each of these two networks has also another equilibrium point which corresponds to the permuted sources up to a scale factor and an additive constant. While the basic network may be unstable at both of its equilibrium points, it is always possible to choose the self-feedback free parameters l_{11} and l_{22} in the extended network to guarantee its stability at one of these points. In particular, in Section 3.4 of [23] we proposed a method for stabilizing the extended network.

The same procedure may be used for computing the equilibrium points of the extended recurrent network and for analyzing its stability when its parameters are matched to the current estimation of the mixing parameters i.e. to $\hat{\boldsymbol{\theta}} = [\hat{L}_{12}, \hat{L}_{21}, \hat{Q}_1, \hat{Q}_2]^T$ in a blind configuration. In the following, we discuss these issues.

3.3.1. Equilibrium points

Consider the extended recurrent network of Fig. 5. For each time n, it performs a recurrence to compute the values of its outputs y_i. We denote by m

the index associated with this recurrence and by $y_i(m)$ the successive values of each output in this recurrence at time n.[7] This recurrence reads

$$\begin{aligned} y_1(m+1) &= x_1(n) + l_{11}y_1(m) + l_{12}y_2(m) + q_1y_1(m)y_2(m), \\ y_2(m+1) &= x_2(n) + l_{21}y_1(m) + l_{22}y_2(m) + q_2y_1(m)y_2(m). \end{aligned} \tag{26}$$

The equilibrium points of this recurrence are all the points (y_1^E, y_2^E) which are such that

$$y_1(m+1) = y_1(m) = y_1^E \quad , \quad y_2(m+1) = y_2(m) = y_2^E. \tag{27}$$

Combining the recurrence (26) with (27) we obtain a system of two equations with respect to the unknowns y_1^E and y_2^E. Solving this system by cancelling y_2^E leads to an algebraic equation of the second degree with respect to y_1^E. Let us define

$$l'_{11} = 1 - l_{11} \quad , \quad l'_{22} = 1 - l_{22} \tag{28}$$

and choose the non-free parameters of the recurrent network as follows

$$l_{12} = \hat{L}_{12}l'_{22} \quad , \quad l_{21} = \hat{L}_{21}l'_{11} \quad , \quad q_1 = \hat{Q}_1 l'_{11}l'_{22} \quad , \quad q_2 = \hat{Q}_2 l'_{11}l'_{22}. \tag{29}$$

Using the same procedure as in Section 3.1 of [23] we can show that the discriminant of the above equation of the second degree reads

$$\hat{\Delta}_{y_1} = (l'_{11}l'_{22})^2\hat{\delta}_{y_1} \tag{30}$$

with

$$\hat{\delta}_{y_1} = [\hat{Q}_2x_1(n) - \hat{Q}_1x_2(n) + \hat{\gamma}]^2 - 4\hat{\alpha}[x_1(n) + x_2(n)\hat{L}_{12}] \tag{31}$$

where

$$\hat{\alpha} = \hat{Q}_2 + \hat{Q}_1\hat{L}_{21}, \tag{32}$$

$$\hat{\gamma} = 1 - \hat{L}_{12}\hat{L}_{21}. \tag{33}$$

[7] These successive ouput values therefore also depend on n. This index n is omitted in the notations $y_i(m)$, in order to improve readability and to focus on the recurrence on outputs for given input values $x_i(n)$.

Note that the above equations are the same as Equations (61)-(64) in [23]: the only difference is that the actual mixing parameters are replaced by their estimates.

The equilibrium points of the recurrent structure are (see Eq. (70) and (71) in [23])

$$y_1^E = \frac{1}{l'_{11}}\hat{s}_1(n) \quad , \quad y_2^E = \frac{1}{l'_{22}}\hat{s}_2(n) \tag{34}$$

and

$$y_1^E = \frac{1}{l'_{11}}\left[\frac{\hat{\beta}}{\hat{\alpha}}\hat{s}_2(n) + \frac{\hat{\gamma}}{\hat{\alpha}}\right] \quad , \quad y_2^E = \frac{1}{l'_{22}}\left[\frac{\hat{\alpha}}{\hat{\beta}}\hat{s}_1(n) - \frac{\hat{\gamma}}{\hat{\beta}}\right] \tag{35}$$

where

$$\hat{\beta} = -(\hat{Q}_2\hat{L}_{12} + \hat{Q}_1). \tag{36}$$

The first equilibrium point (34) provides the signals $\hat{s}_i(n)$ without permutation (and with scale factors) whereas the second point (35) provides these signals with a permutation (and with scale factors and additive constants). This issue is related to the non-bijectivity of the mixing model in $\mathbb{R}^2$ mentioned in Section 3.1.

3.3.2. Stability condition

Once more, the results obtained in [23] about the local stability condition of the network at its equilibrium points for the actual value of the parameters may be generalized to the blind case (i.e. when the network parameters are defined by (28) and (29)) just by replacing the actual parameters by their estimates. Thus, we can propose the following procedure for choosing the free parameters l'_{11} and l'_{22}, which guarantees the local convergence of the extended recurrent network towards an equilibrium point (see Section 3.4 of [23] for details):

1. Choose $\lambda = 1$ or $\lambda = -1$.
2. Set

$$l'_{11} = \mu\frac{\hat{A} + \hat{B}\lambda}{\sqrt{\hat{\delta}_{y_1}}} \quad \text{with} \;\; 0 < \mu < \mu_{max} \tag{37}$$

with

$$\mu_{max} = \begin{cases} 1 & \text{if} \;\; \hat{C}(\lambda) \leq 4 \\ 1 - \sqrt{1 - \frac{4}{\hat{C}(\lambda)}} & \text{otherwise} \end{cases} \tag{38}$$

where

$$\hat{C}(\lambda) = \frac{(\hat{A} + \hat{B}\lambda)^2}{\sqrt{\hat{\delta}_{y_1}}}. \tag{39}$$

3. Set

$$l'_{22} = \lambda l'_{11}. \tag{40}$$

4. Set the parameters of the extended recurrent network according to (28) and (29).

3.3.3. Discussion

Here, we discuss the practical problems which must be considered when using the recurrent structure in a blind configuration.

1. **Negativity of the discriminant:** Until here, we supposed that the estimated parameters, $\hat{\boldsymbol{\theta}}$, took values guaranteeing the bijectivity of the model (23) in the variation domain of sources so that the discriminant (30) was never negative. In practice, however, during the gradient descent algorithm, the discriminant corresponding to the current estimated values of the parameters may become negative for some samples of the observed signals. A first solution may be to reject from the learning database these samples (only for the current update) and to continue the learning procedure with the other samples (if there are enough). Another solution consists in stopping the gradient algorithm and restarting it using other initial values for the estimated mixing parameters. Our experiments show that the second solution leads to better results. Note that this problem is inherent to the non-bijectivity of the mixing model and exists even if we use another method for inverting this model. The following problem, however, is related to the use of recurrent networks.
2. **Global stability:** While the procedure proposed in Section 3.3.2 guarantees the *local stability* of the recurrent network, it cannot guarantee its *global* stability. In fact, when performing the recurrence (26), we must choose an initial value $y_i(0)$ for the network outputs. When choosing the free parameters l'_{11} and l'_{22} as proposed in Section 3.3.2, if the initial value

$y_i(0)$ is around the values of the *unknown* signals $\hat{s}_i(n)$ (divided by l'_{ii}: see Eq. (34)), the network converges towards $\hat{s}_i(n)$ (divided by l'_{ii}). However, initializing the outputs with arbitrary values may lead to divergence. Our experiments show that the stability region around the equilibrium point is usually large and initializing the network outputs with the observed values $x_i(n)$ (divided by l'_{ii}) usually leads to convergence, if the observations have the same order of magnitude as the signals $\hat{s}_i(n)$. However, if by using this initialization the network diverges only for some samples and converges for the others, we can save the convergent samples and use another initialization for the non-convergent samples. If after a fixed number of tests, there still exist non-convergent values, we may reject them from the learning database and continue the learning procedure with the other samples (only in the current iteration of the gradient algorithm).

A better idea consists in initializing the network outputs using the estimation of the sources derived from observations by supposing a linear mixing model. In this approach, we first suppose that the observations are generated using the following linear model

$$\begin{aligned} x_1 &= s_1 - L_{12}s_2, \\ x_2 &= s_2 - L_{21}s_1. \end{aligned} \tag{41}$$

Then, we use a linear classical BSS method to obtain a first estimate of the sources and the mixing parameters L_{12} and L_{21}. The recurrent network outputs y_i may be initialized around these source estimates (divided by l'_{ii}). Moreover, these estimates of the linear mixing parameters may be used for initializing their values in the gradient algorithm. This method gives good results especially when the quadratic part in (16) is small compared to the linear part.

3.4. Algorithm

Thus, we propose the following algorithm for blindly separating linear-quadratic mixtures:

A: Initialize $\hat{\boldsymbol{\theta}} = [\hat{L}_{12}, \hat{L}_{21}, \hat{Q}_1, \hat{Q}_2]^T$ to $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$.

repeat

for $n = 1$ to K **do**

- Compute $\hat{\delta}_{y_1}(n)$ using (31) for the observed samples $x_1(n)$ and $x_2(n)$.
- If $\hat{\delta}_{y_1} < 0$, then go to **A** choosing another value for $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$.
- Compute $l'_{11}(n)$ and $l'_{22}(n)$ from (37) and (40).
- Compute the network parameters $l_{11}(n), l_{22}(n), l_{12}(n), l_{21}(n), q_1(n), q_2(n)$ from (28) and (29).

B: Initialize the outputs of the recurrent network $y_1(n)$ and $y_2(n)$ to the initial values $y_1(0, n)$ and $y_2(0, n)$.

repeat

- Set $y_{1_{new}}(n) = x_1(n) + l_{11}(n)y_1(n) + l_{12}(n)y_2(n) + q_1(n)y_1(n)y_2(n)$.
- Set $y_{2_{new}}(n) = x_2(n) + l_{21}(n)y_1(n) + l_{22}(n)y_2(n) + q_2(n)y_1(n)y_2(n)$.
- If divergence, then go to **B** choosing other values for $y_1(0, n)$ and $y_2(0, n)$.
- Set $y_1(n) = y_{1_{new}}(n)$, $y_2(n) = y_{2_{new}}(n)$.

until the convergence of the recurrence

- Set $\hat{s}_1(n) = l'_{11}(n)y_1(n)$, $\hat{s}_2(n) = l'_{22}(n)y_2(n)$.

end for

- Estimate the score functions of $\hat{s}_1$ and $\hat{s}_2$.
- Compute the gradient $\frac{\partial C}{\partial \hat{\boldsymbol{\theta}}}$ using (25).
- Update the parameter vector using (13).

until the convergence of the parameters

- Set the estimated sources $\tilde{s}_1$ and $\tilde{s}_2$ to $\hat{s}_1$ and $\hat{s}_2$.

4. Simulation results

In Sections 4.1-4.2, we present some simulation results related to the 2×2 linear-quadratic mixture discussed in Section 3. Another test, related to a more general 3×3 linear-quadratic mixture, is presented in Section 4.3.

4.1. Simulations using generalized Gaussian sources and the basic recurrent network

We generated K=1000 samples of two zero-mean, unit-variance, generalized Gaussian sources with pdf $f(s_i) = \frac{\alpha}{2\beta\Gamma(1/\alpha)} \exp(-(\frac{|s_i|}{\beta})^\alpha)$ where the parameter α was varied in the interval $[0.5, 40]$. Note that $\alpha = 1$ corresponds to a Laplace distribution, $\alpha = 2$ to a normal distribution and $\alpha = 40$ to a nearly uniform distribution. The generated sources were then normalized so that all 1000 samples belong to $[-0.5, 0.5]$. This choice guarantees the bijectivity of the mixture in this domain when the mixing parameters are set to the values chosen for this simulation i.e. $L_{12} = -0.2$, $L_{21} = 0.2$, $Q_1 = -0.8$ and $Q_2 = 0.8$.

We used the algorithm of Section 3.4 for separating this mixture. Since the convergence of the extended network was rather slow, we used the basic network in this simulation (i.e. we chose $l_{11} = l_{22} = 0$ in (26)). The basic network is locally stable at the separating point when using the above-mentioned values for sources and mixing parameters.

For each value of α, we performed 100 Monte Carlo simulations corresponding to 100 different initial values of the random source generator. Moreover, we used in each simulation a different initial value, chosen randomly on [-0.05, 0.05], for each entry of $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$. The initial values $y_i(0, n)$ of the network outputs were set to the observed samples $x_i(n)$. For each simulation, the Signal to Interference Ratio (SIR) and the Signal to Interference Ratio Improvement (SIRI) were computed by

$$SIR = \frac{1}{2} \sum_{i=1}^{2} 10 \log_{10} \frac{E_K[s_i^2]}{E_K[(\tilde{s}_i - s_i)^2]} \tag{42}$$

$$SIRI = \frac{1}{2} \sum_{i=1}^{2} 10 \log_{10} \frac{E_K[(x_i - s_i)^2]}{E_K[(\tilde{s}_i - s_i)^2]} \tag{43}$$

after normalizing all the signals so that they have unit variance.

For some values of α, the algorithm diverged for a few number of these simulations. In this first experiment, we simply chose not to take into account these simulations for computing SIR and SIRI, although it was possible to make the

network convergent using the methods explained in the previous section. Figure 6 shows the mean of SIR and SIRI and the standard deviation of SIR, computed on convergent simulations, as a function of α. It can be seen that for these mixing parameters, the algorithm cannot separate Gaussian sources while it succeeds in separating sub-Gaussian and super-Gaussian signals. The average running time of the algorithm implemented in Matlab on a 2.2 GHz AMD PC was 0.45 seconds for each simulation. The scatter plots of the observations and of the estimated sources (for $\alpha = 40$) are shown in Figure 7. As can be seen, the independent components are retrieved by the algorithm.

To determine whether the use of recurrent networks leads to a loss of performance, we also repeated the experiment (for $\alpha = 40$) using the second inversion formula in Eq. (17) for directly calculating the estimated sources at each step of the gradient algorithm. We obtained exactly the same results for SIR and SIRI as when using a recurrent network. This is not surprising because as mentioned in [23], the recurrent network provides very precise estimates.

4.2. Simulation using extended recurrent network

In this simulation, the mixing parameters and the source values were chosen such that:

- the basic recurrent network is unstable at the separating point,
- the Jacobian of the mixing model is negative for all the source samples. As a result, the mixture is bijective in the variation domain of the sources and the permutation problem mentioned in Section 3.1 does not occur at the separating point.

Thus, we chose $L_{12} = L_{21} = -2.5$ and $Q_1 = Q_2 = 1.5$. The two 1000-sample sources were uniformly distributed over [-0.5,0.5]. Since the basic network is unstable and diverges, we only use the extended network in this experiment. We made 100 Monte Carlo simulations corresponding to 100 different initial values of the random source generator. The parameters $\hat{L}_{12}$ and $\hat{L}_{21}$ as well as the network outputs $y_i(n)$ were initialized using the estimates of the parameters

L_{12} and L_{21} and of the sources supposing the linear mixing model (41) for the observations as mentioned in Section 3.3.3. To this end, we used a linear BSS ML algorithm like that proposed in [25]. The parameters $\hat{Q}_1$ and $\hat{Q}_2$ were randomly initialized over [-0.005, 0.005] with a different initial value at each Monte Carlo simulation. The results are shown in the first row of Table 1 and confirm the good performance of our algorithm. Figure 8 shows the evolution of the estimated parameters $\hat{Q}_1$ and $\hat{Q}_2$ during the gradient descent algorithm for one of the 100 simulations. Figure 9 shows the global stability region in this experiment for the source sample $(s_1(n), s_2(n)) = (0.1, -0.1)$ when the estimated mixing parameters are set to their actual values and the parameters l'_{11} and l'_{22} are computed using the procedure proposed in Section 3.3.2, then used for computing the extended network weights using (28) and (29). The stars in this figure represent the initial values $y_1(0)$ and $y_2(0)$ for which the recurrence (26) converges. The circle represents the equilibrium point $(\frac{s_1}{l'_{11}}, \frac{s_2}{l'_{22}})$.

We also repeated the experiment using two Laplacian sources. The results are given in the second row of Table 1.

4.3. Simulations using a linear-quadratic model with 3 sources and 3 mixtures

The illustrative example studied in the first part of this paper and up to this point of the current paper was a linear-quadratic model with 2 sources and 2 mixtures. As explained in the beginning of Section 3.3, in this special case, the use of a recurrent network is not mandatory because a direct separating structure, defined by any of equations (17) may be used instead. In this section, we present our simulation results with a more general model, i.e. a linear-quadratic model with 3 sources and 3 mixtures, whose inverse cannot directly be computed as explained below. The mixing model is defined by:

$$\begin{aligned}
x_1(n) &= s_1(n) + \theta_1 s_2(n) + \theta_2 s_3(n) + \theta_7 s_1(n)s_2(n) + \theta_8 s_1(n)s_3(n) + \theta_9 s_2(n)s_3(n) \\
x_2(n) &= \theta_3 s_1(n) + s_2(n) + \theta_4 s_3(n) + \theta_{10} s_1(n)s_2(n) + \theta_{11} s_1(n)s_3(n) + \theta_{12} s_2(n)s_3(n) \\
x_3(n) &= \theta_5 s_1(n) + \theta_6 s_2(n) + s_3(n) + \theta_{13} s_1(n)s_2(n) + \theta_{14} s_1(n)s_3(n) + \theta_{15} s_2(n)s_3(n)
\end{aligned} \tag{44}$$

where $s_1(n)$, $s_2(n)$ and $s_3(n)$ are three independent sources, $\theta_1, \cdots, \theta_6$ are the mixing parameters related to the linear contributions of the sources in the mixture, while $\theta_7, \cdots, \theta_{15}$ are related to the quadratic contributions. If we compute, from the first equation, s_1 as a function of s_2 and s_3 and replace it in the other two equations, then compute s_2 as a function of s_3 from the second transformed equation and replace it in the third one, we finally obtain an equation which only depends on s_3, the observations x_1, x_2, x_3 and the mixing parameters θ_i. It can be verified that this equation includes the terms containing s_3^5, s_3^4, s_3^3, s_3^2, s_3 and the square roots of some polynomial functions of s_3. It is clear that such an equation cannot analytically be solved for s_3, so that a direct structure cannot be derived. On the contrary, we can invert this mixture using a basic recurrent structure realizing the following recurrence where $\hat{\theta}_i$ represent the estimates of θ_i:

$$\begin{aligned}
y_1(m+1) &= x_1 - \hat{\theta}_1 y_2(m) - \hat{\theta}_2 y_3(m) - \hat{\theta}_7 y_1(m) y_2(m) - \hat{\theta}_8 y_1(m) y_3(m) - \hat{\theta}_9 y_2(m) y_3(m) \\
y_2(m+1) &= x_2 - \hat{\theta}_3 y_1(m) - \hat{\theta}_4 y_3(m) - \hat{\theta}_{10} y_1(m) y_2(m) - \hat{\theta}_{11} y_1(m) y_3(m) - \hat{\theta}_{12} y_2(m) y_3(m) \\
y_3(m+1) &= x_3 - \hat{\theta}_5 y_1(m) - \hat{\theta}_6 y_2(m) - \hat{\theta}_{13} y_1(m) y_2(m) - \hat{\theta}_{14} y_1(m) y_3(m) - \hat{\theta}_{15} y_2(m) y_3(m).
\end{aligned} \tag{45}$$

Appendix B shows how to compute the gradient of the log-likelihood with respect to the parameters θ_i in the mixing model (44), without requiring the knowledge of the explicit inverse of this model. This gradient will then be used to estimate the parameters of the mixing model and to invert this model using the recurrent network (45) as explained in the previous sections.

In our simulation, we chose the following numerical values for the parameters θ_i in (44):

$$\boldsymbol{\theta} = [0.6, 0.5, -0.4, 0.4, 0.3, -0.5, 0.6, 0.4, -0.2, -0.6, 0.4, -0.4, 0.3, -0.3, -0.3]^T.$$

The sources $s_1(n)$, $s_2(n)$ and $s_3(n)$ were three 1000-sample i.i.d. statistically independent signals, uniformly distributed over [-0.5,0.5]. It can be numerically verified that the Jacobian matrix of (44) for this choice of parameters is a P-

matrix[8] on the variation domain of the sources so that the mixture is bijective on this domain [37].

Like in Section 4.2, the mixing parameters corresponding to the linear contribution of the sources in the mixture (i.e. the first 6 estimated parameters $\hat{\theta}_1, \cdots, \hat{\theta}_6$) as well as the recurrent network outputs $y_i(n)$ were initialized using the estimates of the parameters $\theta_1, \cdots, \theta_6$ and of the sources supposing a hypothesized linear mixing model for the observations. The other parameters $\hat{\theta}_7, \cdots, \hat{\theta}_{15}$ were randomly initialized over [-0.05,0.05].

At each step of the gradient algorithm, the parameters $\hat{\theta}_1, \cdots, \hat{\theta}_{15}$ were updated using the gradient computed above[9], then the recurrence (45) was realized until convergence. It can be numerically checked that all the eigenvalues of the Jacobian matrix of (45) are in the unit circle at the separating point so that the network is locally stable at this point. 50 Monte Carlo simulations corresponding to 50 different initial values of random number generator led to mean(SIR)=29.7 dB and std(SIR)=2.5 dB. The mean and the standard deviation of the estimated parameters are given in Table 2. These results confirm the very good performance of our proposed method in a situation where the direct inverse of the mixing model cannot be computed.

5. Conclusion

In this second part of our paper, we proposed a maximum likelihood approach for blindly separating a large class of nonlinear mixtures (called ETM) that we had proposed in the first part of the paper. While this approach exploits the knowledge of the parametric form of the mixing model in the separation procedure, its implementation does not require the knowledge of the *explicit inverse* of the model. Hence, it can be used e.g. for separating polynomial mixtures when the inverse function is unknown.

[8] A matrix $\mathcal{J}$ is called a P-matrix if every principal minor of $\mathcal{J}$ is positive.

[9] We also used a momentum term to improve the convergence of the gradient algorithm.

In particular, we illustrated in detail the proposed approach for a linear-quadratic mixture by using an extended recurrent network with self-feedback parameters which guarantee its local stability. Simulation results showed the very good performance of the proposed algorithm.

Since our approach takes advantage of the knowledge of the parametric form of the mixing model, it can be considered as a structurally constrained ICA method. It is well known [1] that the indeterminacies involved in such methods are much less restrictive than those resulting from general ICA methods which do not exploit this knowledge. However, it is necessary to study the separability issue for each specific mixing model to determine the source distributions for which ICA leads to a unique solution (up to classical indeterminacies). This study for the specific case of linear-quadratic mixtures has been done in the over-determined case [38] and in the determined case with bounded sources [19].

In [23] and this paper, we only studied the stability of the recurrent network in the case of linear-quadratic mixtures with two sources and two observations. When using other kinds of ETM models, such studies are necessary but may be very difficult, which could limit the use of our proposed approach. However, the nonlinear BSS being a difficult problem, the solutions proposed in this paper may help other researchers in their future investigations. Moreover, these solutions may work in practice even without such stability analysis as shown by our simulation results with a 3×3 linear-quadratic model of Section 4.3.

Acknowledgment

The authors would like to thank C. Chaouchi for helpful discussions about the maximum likelihood approach.

A Details of gradient computation for a 2×2 linear-quadratic model

Considering (23), which can be written in vector form $\mathbf{x}(n) = \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})$ and using the notations defined after (12), we obtain

$$\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \hat{\mathbf{s}}} = \begin{pmatrix} 1-\hat{Q}_1\hat{s}_2 & -\hat{L}_{21}-\hat{Q}_2\hat{s}_2 \\ -\hat{L}_{12}-\hat{Q}_1\hat{s}_1 & 1-\hat{Q}_2\hat{s}_1 \end{pmatrix} \tag{A-1}$$

and

$$\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \hat{\boldsymbol{\theta}}}_{|_{\hat{\mathbf{s}}\ cst}} = \begin{pmatrix} -\hat{s}_2 & 0 \\ 0 & -\hat{s}_1 \\ -\hat{s}_1\hat{s}_2 & 0 \\ 0 & -\hat{s}_1\hat{s}_2 \end{pmatrix} \tag{A-2}$$

which implies, from (12)

$$\frac{\partial \hat{\mathbf{s}}}{\partial \hat{\boldsymbol{\theta}}} = \frac{-1}{\hat{J}} \begin{pmatrix} -\hat{s}_2 & 0 \\ 0 & -\hat{s}_1 \\ -\hat{s}_1\hat{s}_2 & 0 \\ 0 & -\hat{s}_1\hat{s}_2 \end{pmatrix} \begin{pmatrix} 1-\hat{Q}_2\hat{s}_1 & \hat{L}_{21}+\hat{Q}_2\hat{s}_2 \\ \hat{L}_{12}+\hat{Q}_1\hat{s}_1 & 1-\hat{Q}_1\hat{s}_2 \end{pmatrix} \tag{A-3}$$

and yields

$$\begin{aligned} \frac{\partial \hat{s}_1}{\partial \hat{\boldsymbol{\theta}}} &= \frac{1}{\hat{J}}\Big[(1-\hat{Q}_2\hat{s}_1)\hat{s}_2\ ,\ (\hat{L}_{12}+\hat{Q}_1\hat{s}_1)\hat{s}_1\ ,(1-\hat{Q}_2\hat{s}_1)\hat{s}_1\hat{s}_2\ ,\ (\hat{L}_{12}+\hat{Q}_1\hat{s}_1)\hat{s}_1\hat{s}_2\Big]^T, \\ \frac{\partial \hat{s}_2}{\partial \hat{\boldsymbol{\theta}}} &= \frac{1}{\hat{J}}\Big[(\hat{L}_{21}+\hat{Q}_2\hat{s}_2)\hat{s}_2\ ,\ (1-\hat{Q}_1\hat{s}_2)\hat{s}_1\ ,(\hat{L}_{21}+\hat{Q}_2\hat{s}_2)\hat{s}_1\hat{s}_2\ ,\ (1-\hat{Q}_1\hat{s}_2)\hat{s}_1\hat{s}_2\Big]^T \end{aligned} \tag{A-4}$$

Using (A-4), we obtain the first term of the gradient (10). To determine the second term, we need to compute $\frac{d\hat{J}}{d\hat{\boldsymbol{\theta}}}$. Denoting $\hat{J} = g(\hat{\boldsymbol{\theta}}, \hat{\mathbf{s}})$ and considering $\hat{\boldsymbol{\theta}}$ as an independent variable and $\hat{\mathbf{s}}$ as a dependent variable, we can write

$$\frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}} = \frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}}_{|_{\hat{\mathbf{s}}\ cst}} + \frac{\partial \hat{\mathbf{s}}}{\partial \hat{\boldsymbol{\theta}}} \frac{\partial \hat{J}}{\partial \hat{\mathbf{s}}}. \tag{A-5}$$

Following (24), we obtain

$$\frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}}_{|_{\hat{\mathbf{s}}\ cst}} = -[\hat{L}_{21}+\hat{Q}_2\hat{s}_2, \hat{L}_{12}+\hat{Q}_1\hat{s}_1, \hat{L}_{21}\hat{s}_1+\hat{s}_2, \hat{s}_1+\hat{L}_{12}\hat{s}_2]^T \tag{A-6}$$

and

$$\frac{\partial \hat{J}}{\partial \hat{\mathbf{s}}} = -[\hat{Q}_2+\hat{L}_{21}\hat{Q}_1, \hat{Q}_1+\hat{L}_{12}\hat{Q}_2]^T. \tag{A-7}$$

Inserting (A-3), (A-6) and (A-7) in (A-5) yields

$$\begin{aligned}\frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}} = [&-(\hat{L}_{21}+\hat{Q}_2\hat{s}_2)-(\hat{Q}_2+\hat{L}_{21}\hat{Q}_1)(1-\hat{Q}_2\hat{s}_1)\hat{s}_2/\hat{J}-(\hat{Q}_1+\hat{L}_{12}\hat{Q}_2)(\hat{L}_{21}+\hat{Q}_2\hat{s}_2)\hat{s}_2/\hat{J},\\ &-(\hat{L}_{12}+\hat{Q}_1\hat{s}_1)-(\hat{Q}_1+\hat{L}_{12}\hat{Q}_2)(1-\hat{Q}_1\hat{s}_2)\hat{s}_1/\hat{J}-(\hat{Q}_2+\hat{L}_{21}\hat{Q}_1)(\hat{L}_{12}+\hat{Q}_1\hat{s}_1)\hat{s}_1/\hat{J},\\ &-(\hat{L}_{21}\hat{s}_1+\hat{s}_2)-(\hat{Q}_2+\hat{L}_{21}\hat{Q}_1)(1-\hat{Q}_2\hat{s}_1)\hat{s}_1\hat{s}_2/\hat{J}-(\hat{Q}_1+\hat{L}_{12}\hat{Q}_2)(\hat{L}_{21}+\hat{Q}_2\hat{s}_2)\hat{s}_1\hat{s}_2/\hat{J},\\ &-(\hat{L}_{12}\hat{s}_2+\hat{s}_1)-(\hat{Q}_1+\hat{L}_{12}\hat{Q}_2)(1-\hat{Q}_1\hat{s}_2)\hat{s}_1\hat{s}_2/\hat{J}-(\hat{Q}_2+\hat{L}_{21}\hat{Q}_1)(\hat{L}_{12}+\hat{Q}_1\hat{s}_1)\hat{s}_1\hat{s}_2/\hat{J}]^T.\end{aligned} \tag{A-8}$$

Using (A-4), (A-8) and (10), we finally obtain the expression (25) for the gradient of the cost function C.

B Details of gradient computation for a 3×3 linear-quadratic model

To compute the gradient of the cost function C defined in (8), we first compute the Jacobian matrix of the mixing model (44) which reads

$$\boldsymbol{\mathcal{J}} = \frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \mathbf{s}(n)} = \begin{pmatrix} 1+\theta_7 s_2(n)+\theta_8 s_3(n) & \theta_3+\theta_{10} s_2(n)+\theta_{11} s_3(n) & \theta_5+\theta_{13} s_2(n)+\theta_{14} s_3(n) \\ \theta_1+\theta_7 s_1(n)+\theta_9 s_3(n) & 1+\theta_{10} s_1(n)+\theta_{12} s_3(n) & \theta_6+\theta_{13} s_1(n)+\theta_{15} s_3(n) \\ \theta_2+\theta_8 s_1(n)+\theta_9 s_2(n) & \theta_4+\theta_{11} s_1(n)+\theta_{12} s_2(n) & 1+\theta_{14} s_1(n)+\theta_{15} s_2(n) \end{pmatrix}. \tag{B-1}$$

Denoting by $\mathcal{J}_{ij}$ the entry (i,j) of this matrix, the Jacobian, *i.e.* the determinant of (B-1), can be written as

$$J = \mathcal{J}_{11}(\mathcal{J}_{22}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{32}) - \mathcal{J}_{12}(\mathcal{J}_{21}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{31}) + \mathcal{J}_{13}(\mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31}). \tag{B-2}$$

Computing the derivative of (44) with respect to $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_{15}]^T$, keeping $s_i(n)$ constant, yields

$$\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n), \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}(n)\ cst}} =$$

$$\begin{pmatrix} s_2 & s_3 & 0 & 0 & 0 & 0 & s_1s_2 & s_1s_3 & s_2s_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s_1 & s_3 & 0 & 0 & 0 & 0 & 0 & s_1s_2 & s_1s_3 & s_2s_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_1 & s_2 & 0 & 0 & 0 & 0 & 0 & 0 & s_1s_2 & s_1s_3 & s_2s_3 \end{pmatrix}^T \tag{B-3}$$

where the time index (n) is omitted for the sake of readability. Replacing s and θ by $\hat{s}$ and $\hat{\theta}$ in (B-1) and (B-3), then using (12), we can compute $\frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\boldsymbol{\theta}}}$. This derivative, together with the conditional score functions, allows us to obtain the first term of the gradient (10). To determine the second term, we need to compute $\frac{\partial J}{\partial \boldsymbol{\theta}}$. Denoting $J = g(\boldsymbol{\theta}, \mathbf{s})$ and considering $\boldsymbol{\theta}$ as an independent variable and $\mathbf{s}$ as a dependent variable, we can write

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \frac{\partial J}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}\ cst}} + \frac{\partial \mathbf{s}}{\partial \boldsymbol{\theta}} \frac{\partial J}{\partial \mathbf{s}}. \tag{B-4}$$

Using (B-1) and (B-2) and after some computation, we obtain

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}\ cst}} = [&\mathcal{J}_{13}\mathcal{J}_{32}-\mathcal{J}_{12}\mathcal{J}_{33}\ ,\ \mathcal{J}_{12}\mathcal{J}_{23}-\mathcal{J}_{13}\mathcal{J}_{22}\ ,\ \mathcal{J}_{23}\mathcal{J}_{31}-\mathcal{J}_{21}\mathcal{J}_{33}\ ,\ \mathcal{J}_{13}\mathcal{J}_{21}-\mathcal{J}_{11}\mathcal{J}_{23}\ ,\ \mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31}\ , \\ &\mathcal{J}_{12}\mathcal{J}_{31}-\mathcal{J}_{11}\mathcal{J}_{32}\ ,\ s_2(\mathcal{J}_{22}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{32})-s_1(\mathcal{J}_{12}\mathcal{J}_{33}-\mathcal{J}_{13}\mathcal{J}_{32})\ ,\ s_3(\mathcal{J}_{22}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{32})+s_1(\mathcal{J}_{12}\mathcal{J}_{23}-\mathcal{J}_{13}\mathcal{J}_{22})\ , \\ &-s_3(\mathcal{J}_{12}\mathcal{J}_{33}-\mathcal{J}_{13}\mathcal{J}_{32})+s_2(\mathcal{J}_{12}\mathcal{J}_{23}-\mathcal{J}_{13}\mathcal{J}_{22})\ ,\ -s_2(\mathcal{J}_{21}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{31})+s_1(\mathcal{J}_{11}\mathcal{J}_{33}-\mathcal{J}_{13}\mathcal{J}_{31})\ , \\ &-s_3(\mathcal{J}_{21}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{31})-s_1(\mathcal{J}_{11}\mathcal{J}_{23}-\mathcal{J}_{13}\mathcal{J}_{21})\ ,\ s_3(\mathcal{J}_{11}\mathcal{J}_{33}-\mathcal{J}_{13}\mathcal{J}_{31})-s_2(\mathcal{J}_{11}\mathcal{J}_{23}-\mathcal{J}_{13}\mathcal{J}_{21})\ , \\ &s_2(\mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31})-s_1(\mathcal{J}_{11}\mathcal{J}_{32}-\mathcal{J}_{12}\mathcal{J}_{31})\ ,\ s_3(\mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31})+s_1(\mathcal{J}_{11}\mathcal{J}_{22}-\mathcal{J}_{12}\mathcal{J}_{21})\ , \\ &-s_3(\mathcal{J}_{11}\mathcal{J}_{32}-\mathcal{J}_{12}\mathcal{J}_{31})+s_2(\mathcal{J}_{11}\mathcal{J}_{22}-\mathcal{J}_{12}\mathcal{J}_{21})]^T \end{aligned} \tag{B-5}$$

and

$$\frac{\partial J}{\partial \mathbf{s}} = [\mathcal{J}_{11}(\theta_{10}\mathcal{J}_{33}+\theta_{14}\mathcal{J}_{22}-\theta_{13}\mathcal{J}_{32}-\theta_{11}\mathcal{J}_{23})-\mathcal{J}_{12}(\theta_7\mathcal{J}_{33}+\theta_{14}\mathcal{J}_{21}-\theta_{13}\mathcal{J}_{31}-\theta_8\mathcal{J}_{23})$$
$$+\mathcal{J}_{13}(\theta_7\mathcal{J}_{32}+\theta_{11}\mathcal{J}_{21}-\theta_{10}\mathcal{J}_{31}-\theta_8\mathcal{J}_{22}),$$
$$\theta_7(\mathcal{J}_{22}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{32})+\mathcal{J}_{11}(\theta_{15}\mathcal{J}_{22}-\theta_{12}\mathcal{J}_{23})-\theta_{10}(\mathcal{J}_{21}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{31})-\mathcal{J}_{12}(\theta_{15}\mathcal{J}_{21}-\theta_9\mathcal{J}_{23})$$
$$+\theta_{13}(\mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31})+\mathcal{J}_{13}(\theta_{12}\mathcal{J}_{21}-\theta_9\mathcal{J}_{22}),$$
$$\theta_8(\mathcal{J}_{22}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{32})+\mathcal{J}_{11}(\theta_{12}\mathcal{J}_{33}-\theta_{15}\mathcal{J}_{32})-\theta_{11}(\mathcal{J}_{21}\mathcal{J}_{33}-\mathcal{J}_{23}\mathcal{J}_{31})-\mathcal{J}_{12}(\theta_9\mathcal{J}_{33}-\theta_{15}\mathcal{J}_{31})$$
$$+\theta_{14}(\mathcal{J}_{21}\mathcal{J}_{32}-\mathcal{J}_{22}\mathcal{J}_{31})+\mathcal{J}_{13}(\theta_9\mathcal{J}_{32}-\theta_{12}\mathcal{J}_{31})]^T. \quad \text{(B-6)}$$

Replacing J, s and θ by $\hat{J}$, $\hat{s}$ and $\hat{\theta}$ and using the expression of $\frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\boldsymbol{\theta}}}$, already computed, we obtain $\frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}}$ from (B-4), then the second term of (10), and therefore the entire gradient $\frac{dC}{d\hat{\boldsymbol{\theta}}}$.

References

[1] P. Comon and C. Jutten Eds., *Handbook of blind source separation. Independent component analysis and applications*. Academic Press, Oxford, 2010.

[2] H. Lappalainen, J. Miskin, Ensemble learning, in: M. Girolami (Ed.), *Advances in Independent Component Analysis*, Springer-Verlag, 2000, pp. 93-121.

[3] Y. Tan, J. Wang, J. Zurada, Nonlinear blind source separation using a radial basis function network, *IEEE Transactions on Neural Networks*, 12 (2001), pp. 124-134.

[4] L. B. Almeida, MISEP - linear and nonlinear ICA based on mutual information, *Journal of Machine Learning Research*, Vol. 4, pp. 1297-1318, 2003.

[5] S. Harmeling, A. Ziehe, B. Blankertz, K-R. Müller, Kernel-based nonlinear blind source separation, *Neural Computation*, 15(2003), pp. 1089-1124.

[6] A. Honkela, H. Valpola, A. Ilin, J. Karhunen, Blind separation of nonlinear mixtures by variational Bayesian learning, *Digital Signal Processing*, 17 (2007), pp. 914-934.

[7] A. Hyvarinen and P. Pajunen, Nonlinear independent component analysis: Existence and uniqueness results, *Neural Networks*, 12(3), pp. 429-439, 1999.

[8] A. Taleb and C. Jutten, Source separation in post-nonlinear mixtures, *IEEE Trans. on Signal Processing*, 47(10), pp. 2807-2820, 1999.

[9] A. Taleb, A generic framework for blind source separation in structured nonlinear models, *IEEE Trans. on Signal Processing*, 50(8), pp. 1819-1830, August 2002.

[10] C. Jutten, B. Babaie-Zadeh, S. Hosseini, Three easy ways for separating nonlinear mixtures?, *Signal Processing*, 84(2), pp. 217-229, Feb. 2004.

[11] M. Babaie-Zadeh, C. Jutten, K. Nayebi, A geometric approach for separating post nonlinear mixtures, in *Proc. of EUSIPCO 2002*, vol. 2, pp. 11-14, Toulouse, Sep. 2002.

[12] S. Achard, D. Pham, C. Jutten, Blind source separation in post nonlinear mixtures, in *Proc. of ICA 2001*, San Diego (California, USA), 2001, pp. 259-300.

[13] M. Krob and M. Benidir, Blind identification of a linear-quadratic model using higher-order statistics, In *Proc. ICASSP*, vol. 4, pp. 440-443, 1993.

[14] K. Abed-Meraim, A. Belouchrani, and Y. Hua, Blind identification of a linear-quadratic mixture of independent components based on joint diagonalization procedure, In *Proc. ICASSP*, pp. 2718-2721, Atlanta, USA, May 1996.

[15] S. Hosseini, Y. Deville, Blind separation of linear-quadratic mixtures of real sources using a recurrent structure, in *Proc. 7th Int. Work-Conf. Artificial*

Neural Networks, IWANN'03, vol. 2, pp. 241-248, Mao, Menorca, Spain, June 2003.

[16] S. Hosseini, Y. Deville, Blind maximum likelihood separation of a linear-quadratic mixture, In *Proc. of ICA'04*, pp. 694-701, Granada, Sep. 2004. (ERRATUM: http://arxiv.org/abs/1001.0863).

[17] M. S. C. Almeida and L. B. Almeida, Separating nonlinear image mixtures using a physical model trained with ICA, In *Proc. 2006 IEEE Int. Worksh. Machine Learning for Signal Processing*, Maynooth, Ireland, September 2006.

[18] M. Castella, Inversion of polynomial systems and separation of nonlinear mixtures of finite-alphabet sources. *IEEE Trans. Signal Processing*, Vol. 56, Issue 8, Part 2, Aug. 2008, pp. 3905-3917.

[19] M. S. C. Almeida and L. B. Almeida, Nonlinear separation of show-through image mixtures using a physical model trained with ICA, *Signal Processing*, vol. 92, issue 4, pp. 872-884, April 2012.

[20] J. Eriksson and V. Koivunen, Blind identifiability of class of nonlinear instantaneous ICA models. In *Proc. of EUSIPCO 2002*, vol. 2, pp. 7-10, Toulouse, Sep. 2002.

[21] G. Bedoya, *Non-linear blind signal separation for chemical solid-state sensor arrays*, PhD Thesis, Technical Univ. of Catalonia, Dept. of Electrical Eng., Barcelona, Spain, 2006.

[22] L. T. Duarte, C. Jutten, Blind Source Separation of a Class of Nonlinear Mixtures, In *Proc. ICA'07*, pp. 41-48, London, Sep. 2007.

[23] Y. Deville, S. Hosseini, Recurrent networks for separating extractable-target nonlinear mixtures. Part I : non-blind configurations, *Signal Processing*, vol. 89, no. 4, pp. 378-393, April 2009.

[24] M. Gaeta and J.-L. Lacoume, Source separation without prior knowledge: the maximum likelihood solution, in *Proc. of EUSIPCO 90*, pp. 621-624, 1990.

[25] D.-T. Pham and P. Garat, Blind separation of mixture of independent sources through a quasi-maximum likelihood approach, *IEEE Trans. on Signal Processing*, 45(7), pp. 1712-1725, 1997.

[26] J.-F. Cardoso, Infomax and maximum likelihood for source separation, *IEEE Signal Processing Letters*, 4 (1997), pp. 112-114.

[27] A. J. Bell and T. J. Sejnowski, A non-linear information maximization approach to blind separation and blind deconvolution, *Neural Computation*, 7 (1995), pp. 1129-1159.

[28] P. Comon, Independent component analysis - a new concept?, *Signal Processing*, 36 (1994), pp. 287-314.

[29] J.-F. Cardoso, Blind signal separation: statistical principles, *Proceedings of the IEEE*, 86 (1998), pp. 2009-2025.

[30] M. Babaie-Zadeh, C. Jutten, A general approach for mutual information minimization and its application to blind source separation, *Signal Processing*, 85 (2005), pp. 975-995.

[31] F. Merrikh-Bayat, M. Babaie-Zadeh and C. Jutten, A nonlinear blind source separation solution for removing the show-through effect in the scanned documents, in *Proc. of EUSIPCO 2008*, Lausanne, Switzerland, August 2008.

[32] L. B. Almeida, Separating a Real-Life Nonlinear Image Mixture, *Journal of Machine Learning Research*, vol. 6, pp. 1199-1229, 2005.

[33] B. Somers, K. Cools, S. Delalieux, J. Stuckens, D. Van der Zande, W.W. Verstraeten and P. Coppin, Nonlinear hyperspectral mixture analysis for

tree cover estimates in orchards, *Remote Sensing of Environment* 113 (2009), pp. 1183-1193.

[34] W. Fan, B. Hu, J. Miller, and M. Li, Comparative study between a new nonlinear model and common linear model for analyzing laboratory simulated forest hyperspectral data, *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951-2962, June 2009.

[35] J. Nascimento and J. Bioucas-Dias, Nonlinear mixture model for hyperspectral unmixing, in *Proceedings of the SPIE Conference on Image and Signal Processing for Remote Sensing*, vol. SPIE-7477, 2009.

[36] I. Meganem, P. Déliot, X. Briottet, Y. Deville, S. Hosseini, Physical modelling and non-linear unmixing method for urban hyperspectral images, in *Proceedings of the 2011 IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2011)*, Lisbon, Portugal, June 2011.

[37] T. Parthasarathy, On global univalence theorems. *Lecture notes in mathematics* 977, Springer-Verlag, 1983.

[38] F.J. Theis and W. Nakamura, Quadratic independent component analysis, IEICE Trans. Fundamentals, E87-A(9):2355-2363, 2004.

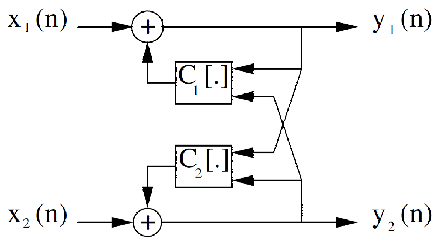

Figure 1: Proposed recurrent network for basic additive-target mixtures.

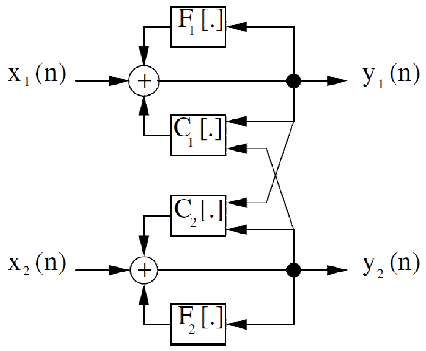

Figure 2: Proposed recurrent network for basic additive-target mixtures, with additional self-feedback.

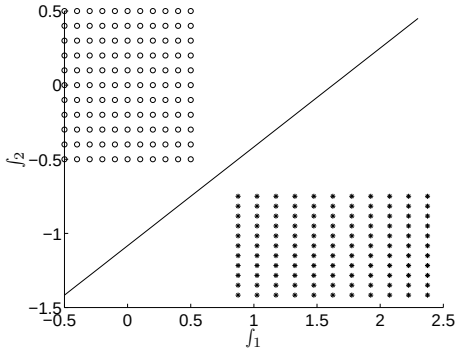

Figure 3: Case when $J > 0$ for all the source values. Circles: $(\int_1, \int_2)_2 = (s_1, s_2)$. Stars: $(\int_1, \int_2)_1 = (-\frac{Q_1+L_{12}Q_2}{Q_2+L_{21}Q_1}s_2 - \frac{L_{12}L_{21}-1}{Q_2+L_{21}Q_1}, -\frac{Q_2+L_{21}Q_1}{Q_1+L_{12}Q_2}s_1 - \frac{L_{12}L_{21}-1}{Q_1+L_{12}Q_2})$. Straight line: $J_1((\int_1, \int_2)) = 0$.

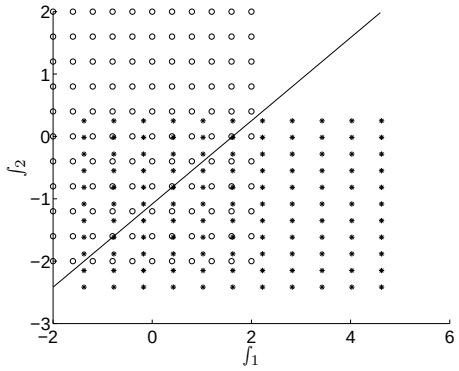

Figure 4: Case when $J > 0$ for some values of the sources and $J < 0$ for other values. Circles: non-permuted source samples. Stars: permuted source samples. Straight line: $J_1((\int_1, \int_2)) = 0$.

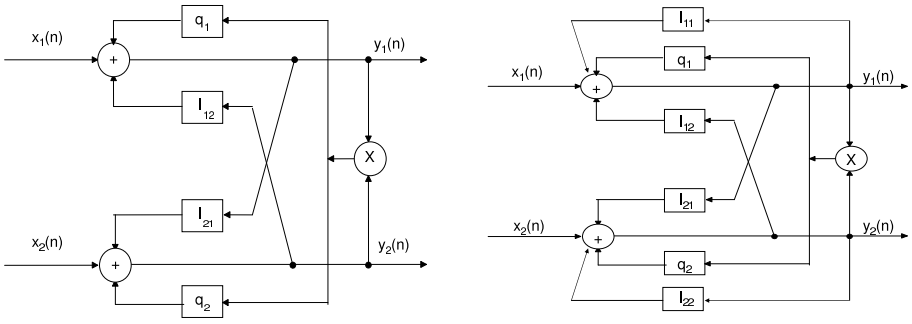

Figure 5: Recurrent networks for separating linear-quadratic mixtures. Left: Basic version (without self-feedback). Right: extended version (with self-feedback).

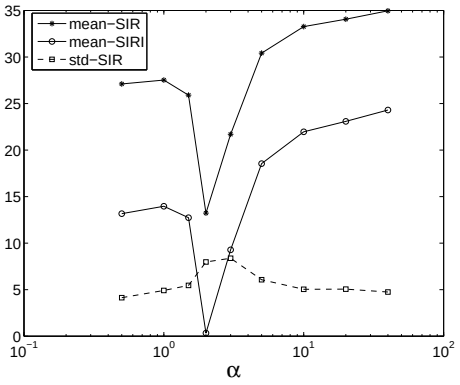

Figure 6: mean of SIR (stars), mean of SIRI (circles), and standard deviation of SIR (squares) as functions of α.

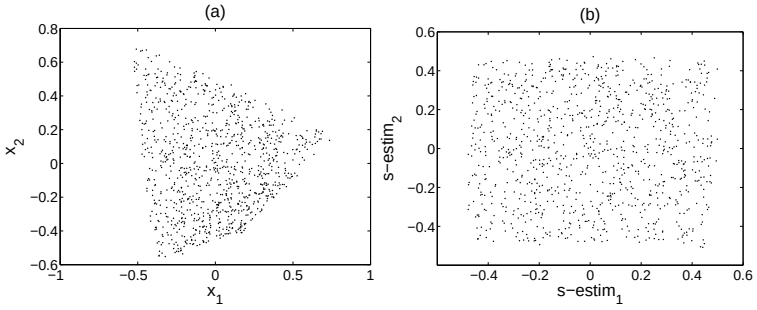

Figure 7: scatter plots of (a) observations, and (b) estimated sources.

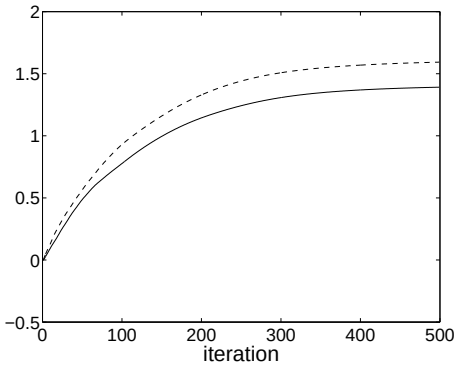

Figure 8: Evolution of the estimated parameters $\hat{Q}_1$ (solid line) and $\hat{Q}_2$ (dashed line) during gradient descent algorithm. The actual values are: $Q_1 = Q_2 = 1.5$.

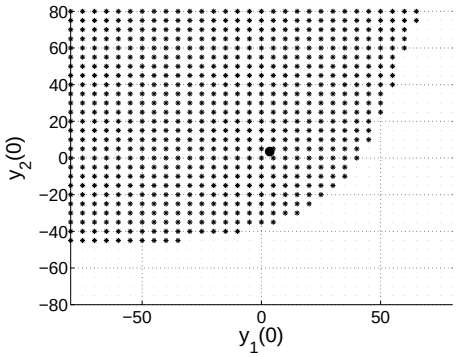

Figure 9: Global stability region (stars) and Equilibrium point (circle) for $(s_1(n), s_2(n)) = (0.1, -0.1)$.

	mean(SIR)	std(SIR)	min(SIR)	max(SIR)	mean(SIRI)
Uniform sources	25.04	1.11	22.60	27.65	25.99
Laplacian sources	21.77	6.37	2.35	32.37	22.82

Table 1: Mean, standard deviation, minimum, and maximum of SIR and mean of SIRI.

i	1	2	3	4	5	6	7	8
θ_i	0.600	0.500	-0.400	0.400	0.300	-0.500	0.600	0.400
mean($\hat{\theta}_i$)	0.596	0.506	-0.397	0.405	0.297	-0.497	0.605	0.405
std($\hat{\theta}_i$)	0.024	0.028	0.034	0.027	0.021	0.018	0.049	0.067
i	9	10	11	12	13	14	15	
θ_i	-0.200	-0.600	0.400	-0.400	0.300	-0.300	-0.300	
mean($\hat{\theta}_i$)	-0.211	-0.599	0.401	-0.400	-0.291	-0.304	-0.288	
std($\hat{\theta}_i$)	0.063	0.036	0.082	0.042	0.056	0.053	0.043	

Table 2: Actual values of the parameters (θ_i), and means and standard deviations of their estimates ($\hat{\theta}_i$).