Blind Separation of Parametric Nonlinear Mixtures of Possibly Autocorrelated and Non-Stationary Sources

Shahram Hosseini and Yannick Deville, *Member, IEEE*

***Abstract*—In this paper, we present a new method, formulated in a maximum-likelihood framework, for blindly separating nonlinear mixtures of statistically independent signals. Our method exploits, on the one hand, the knowledge of the parametric model of the mixing transformation (with unknown parameter values), and on the other hand, the possible structure of source signals, i.e., their autocorrelation and/or nonstationarity. One of the main advantages of the proposed method is that it can be implemented even if the analytical expression of the inverse model is unknown. The method is first addressed in a general configuration, then detailed for two special cases, i.e., a simple bijective "toy" model and a linear-quadratic model. The study of the toy model is interesting because of its simplicity and its global bijectivity, which allows us to focus our efforts on parameter estimation. The linear-quadratic model is chosen due to its capacity to describe real-world mixing phenomena. Simulation results, using the toy model and using a subclass of the linear-quadratic model (i.e., the bilinear model), show that taking into account the nonlinearity of the mixing transformations and the structure of signals considerably improves separation performance.**

***Index Terms*—Autocorrelation, blind source separation, independent component analysis, maximum likelihood, non-stationarity, nonlinear mixtures.**

I. Introduction

DURING the last three decades, linear Blind Source Separation (BSS), which aims to separate source signals from their observed linear mixtures, has been largely studied (see for example the handbook [1]). The main approach to realize BSS is based on Independent Component Analysis (ICA) and the related methods which exploit three different features: non-Gaussianity, temporal autocorrelation or non-stationarity of the sources [2]. In [3] and [4], we showed how these three features may be used together to improve linear BSS performance. Nonlinear BSS is a less studied and more difficult subject. Several algorithms have been proposed for separating general nonlinear mixtures (see e.g., [5]–[9]) but they are not devoted to a specific class of mixtures and do not take advantage of possible knowledge of parametric mixing models. It is however well known that the independence assumption is not enough for separating general nonlinear mixtures because of the non-trivial indeterminacies leading to the non-uniqueness of the solution of nonlinear ICA [10], [11]. To reduce these indeterminacies, one may constrain the structure of mixing models [12], [13]. Thus, several specific classes of nonlinear mixtures have been studied in the literature, e.g., post-nonlinear mixtures [11], [14], [15], linear-quadratic mixtures [16]–[21], linearizable mappings [22] or nonlinear mixtures encountered with gas or chemical sensors [23], [24] or with quantum sources [25] (see also [12] for a more general framework and Chapter 14 of [1] for more references). Nevertheless, these works do not exploit the possible autocorrelation or non-stationarity of the signals in the separating procedure.

An attempt to use time correlation for separating nonlinear mixtures can be found in [8]. In that work, data are first mapped from input space to a kernel feature space, then the dimensionality is reduced and a second-order temporal decorrelation BSS algorithm is used, finally an automatic selection procedure is applied to recover the sources. In another work [26], the authors propose to exploit time correlation by combining second-order ICA and slow feature analysis which aims at finding a representation where signal components are slowly varying. The slowness is measured using the variance of the first derivative. In [27] and [28], the authors exploit time correlation in the separation of post-nonlinear mixtures. In [27] an alternating conditional expectation algorithm is applied to approximately invert the post-nonlinear function, then a temporal decorrelation algorithm is used to recover the source signals. In [28], a similar method is proposed which replaces the first stage by a Gaussianizing transformation. The motivation is that linearly mixed signals before nonlinear transformation are approximately Gaussian-distributed.

Even though these methods make use of time correlation, they either do not exploit the known structure of the mixing model or are limited to the special case of post-nonlinear mixtures. In this paper, we propose a new maximum likelihood (ML) approach for taking into account the autocorrelation and non-stationarity of the sources to achieve nonlinear BSS, supposing that the parametric mixing model is known (but its parameter values are unknown)[1]. The ML approach for *linear* BSS was initially proposed in [29] and was extended in [30] and [31]. It is closely related to the Mutual Information approach which

Manuscript received April 04, 2014; revised July 26, 2014 and October 13, 2014; accepted October 22, 2014. Date of publication November 04, 2014; date of current version November 14, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andre L. F. Almeida.

The authors are with Institut de Recherche en Astrophysique et Planétologie (IRAP), Université de Toulouse, CNRS, 31400 Toulouse, France (e-mail: shosseini@irap.omp.eu; Shahram.Hosseini@irap.omp.eu; ydeville@irap.omp.eu; Yannick.Deville@irap.omp.eu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2367474

[1]Since the parametric model of the mixing transformation is known except for some unknown parameters, some authors propose to call this type of methods *semi-blind*. However, according to this argument, linear source separation methods based on ICA should also be called semi-blind because these methods also assume that the parametric model of the mixing transformation is known.

has been used for separating linear [32], [33] and nonlinear [7], [12], [34] mixtures in the case of independent and identically distributed (i.i.d.) signals. We also proposed an ML approach for separating linear-quadratic mixtures of i.i.d. sources using recurrent networks [19], [35].

The main advantages of the approach proposed in the current paper are:

- Exploiting the knowledge of the parametric model of mixing transformation permits one to constrain the initially highly ill-posed problem.
- An original use of implicit differentiation allows one to derive the analytical expression of the gradient of the considered cost function without requiring the knowledge of the explicit inverse of the mixing model.
- Exploiting the structure of source signals (autocorrelation, non-stationarity) in the ML criterion leads to better performance.

The remainder of this paper is organized as follows. Our ML method is explained in Section II. In Section III, we illustrate the proposed procedure using a simple bijective toy model. Section IV presents the theoretical analysis of our method for the linear-quadratic model and its experimental validation using a subclass of this model, called the bilinear model. We finally conclude in Section V.

II. Method

We consider the parametric mixing model

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

where $\mathbf{s}(n) = [s_1(n), \ldots, s_K(n)]^T$ is the vector of K independent unknown source signals at time n (^{T} stands for transposition), $\mathbf{x}(n) = [x_1(n), \ldots, x_P(n)]^T$ is the vector of P observed signals and $\boldsymbol{\mathcal{F}} = [\mathcal{F}_1, \ldots, \mathcal{F}_P]^T$ is a memoryless parametric function, defined by the unknown parameter vector $\boldsymbol{\theta}$, which is supposed to be differentiable with respect to $\mathbf{s}(n)$ and $\boldsymbol{\theta}$.[2] In the following, we only consider the determined case where $P = K$, because our method is only applicable to determined mixtures. Note that an over-determined mixture ($P > K$) may be reduced to a determined one by simply ignoring $P-K$ observed signals. However, to take into account all available data, a better solution may consist in gathering observations in groups of K members, then applying our method to estimate a mixing parameter vector for each group, and finally choosing a single parameter vector from all these estimates, e.g., by computing their mean or median.

[2]Note that our analysis does not hold if this differentiability assumption is violated (e.g., if $\boldsymbol{\theta}$ takes discrete values).

A. Cost Function

Suppose we are given N samples of each observed signal corresponding to $n = 1, 2, \ldots, N$, and denote by $f_{S_i}^{(N)}(.)$ the joint probability density function (pdf) of N corresponding samples of each source s_i. The joint pdf of all K independent source signals reads $f_{\mathbf{S}}^{(N)}(.) = \prod_{i=1}^{K} f_{S_i}^{(N)}(.)$. Let $\hat{\boldsymbol{\Theta}}$ be 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 the mixing model (1) for each vector $\hat{\boldsymbol{\theta}} \in \hat{\boldsymbol{\Theta}}$ will be denoted by $\hat{\mathbf{s}}(n) = \boldsymbol{\mathcal{F}}^{-\mathbf{1}}(\mathbf{x}(n), \hat{\boldsymbol{\theta}})$.

The source pdf $f_{\mathbf{S}}^{(N)}(.)$ being fixed, the distribution of the transformed vector $\boldsymbol{\mathcal{F}}(\mathbf{s}, \hat{\boldsymbol{\theta}})$ only depends on $\hat{\boldsymbol{\theta}}$. This family of distributions is used as a parametric model for the pdf of observations[3] and is denoted by $\mathcal{P} = \{p_{\hat{\boldsymbol{\theta}}}\ ,\ \hat{\boldsymbol{\theta}} \in \hat{\boldsymbol{\Theta}}\}$.

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

$$L = p_{\hat{\boldsymbol{\theta}}}(x_1(1), \ldots, x_K(1), \ldots\ldots, x_1(N), \ldots, x_K(N)). \tag{2}$$

Let's denote by $\boldsymbol{\mathcal{J}_N}$ the $KN \times KN$ Jacobian matrix of the global transformation relating N samples of K sources to N samples of K mixtures, defined by (3) at the bottom of the page. We also denote by $J_N = det\boldsymbol{\mathcal{J}_N}$ the Jacobian of this global transformation, and by $\hat{J}_N$ its value evaluated at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and $\mathbf{s} = \hat{\boldsymbol{s}}$. Then, thanks to the assumed bijectivity of the mixture, we can write

$$\begin{aligned} L &= \frac{f_{\mathbf{S}}^{(N)}(\hat{s}_1(1), \ldots, \hat{s}_K(1), \ldots\ldots, \hat{s}_1(N), \ldots, \hat{s}_K(N))}{|\hat{J}_N|} \\ &= \frac{\prod_{i=1}^{K} f_{S_i}^{(N)}(\hat{s}_i(1), \ldots, \hat{s}_i(N))}{|\hat{J}_N|}. \end{aligned} \tag{4}$$

Since the mixing transformation (1) is memoryless, the Jacobian matrix $\boldsymbol{\mathcal{J}_N}$ is block-diagonal. As a result, the Jacobian J_N can be written as

$$J_N = \prod_{n=1}^{N} J(n), \tag{5}$$

[3]This parametric pdf is equal to the actual observation pdf if $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$.

$$\boldsymbol{\mathcal{J}_N} = \begin{pmatrix} \frac{\partial \mathcal{F}_1(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_1(1)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_1(1)} & \ldots & \ldots & \frac{\partial \mathcal{F}_1(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_1(1)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_1(1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \frac{\partial \mathcal{F}_1(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_K(1)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_K(1)} & \ldots & \ldots & \frac{\partial \mathcal{F}_1(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_K(1)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_K(1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \frac{\partial \mathcal{F}_1(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_1(N)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_1(N)} & \ldots & \ldots & \frac{\partial \mathcal{F}_1(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_1(N)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_1(N)} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \frac{\partial \mathcal{F}_1(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_K(N)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(1),\boldsymbol{\theta})}{\partial s_K(N)} & \ldots & \ldots & \frac{\partial \mathcal{F}_1(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_K(N)} & \ldots & \frac{\partial \mathcal{F}_K(\mathbf{s}(N),\boldsymbol{\theta})}{\partial s_K(N)} \end{pmatrix} \tag{3}$$

where $J(n)$ is the Jacobian of the mixing transformation (1) at time n, i.e., the determinant of the $K \times K$ Jacobian matrix $\boldsymbol{\mathcal{J}}(n) = \frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \mathbf{s}(n)}$ with the generic entry $\mathcal{J}_{ij}(n) = \frac{\partial \mathcal{F}_j(\mathbf{s}(n),\boldsymbol{\theta})}{\partial s_i(n)}$. We also denote by $\hat{J}(n)$ the value of $J(n)$ evaluated at $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{s}}(n)$.

The maximum likelihood estimate of the actual parameter vector $\boldsymbol{\theta}$ is obtained by maximizing the likelihood (4) with respect to $\hat{\boldsymbol{\theta}}$. Equivalently, we can minimize the cost function $C = \frac{-1}{N} \log L$ which can be rewritten using (5) as

$$C = \frac{-1}{N} \sum_{i=1}^{K} \log f_{S_i}^{(N)}(\hat{s}_i(1), \ldots, \hat{s}_i(N)) + \frac{1}{N} \sum_{n=1}^{N} \log |\hat{J}(n)|. \tag{6}$$

Depending on the possible autocorrelation and/or non-stationarity of the sources, four different cases may be considered:

1) Each source is an i.i.d. signal (i.e., stationary and with independent samples). In this case, we can write $f_{S_i}^{(N)}(\hat{s}_i(1), \ldots, \hat{s}_i(N)) = \prod_{n=1}^{N} f_{S_i}(\hat{s}_i(n))$, where $f_{S_i}(.)$ is the marginal pdf of the source s_i which does not depend on n. Thus, the cost function (6) can be rewritten as

$$C = \left(\frac{-1}{N} \sum_{i=1}^{K} \sum_{n=1}^{N} \log f_{S_i}(\hat{s}_i(n)) \right) + \frac{1}{N} \sum_{n=1}^{N} \log |\hat{J}(n)|. \tag{7}$$

2) Each source has independent samples but is possibly non-stationary. In this case, we have

$$C = \left(\frac{-1}{N} \sum_{i=1}^{K} \sum_{n=1}^{N} \log f_{S_i(n)}(\hat{s}_i(n)) \right) + \frac{1}{N} \sum_{n=1}^{N} \log |\hat{J}(n)|, \tag{8}$$

where $f_{S_i(n)}(.)$ is the marginal pdf of source s_i at time n, which depends on n. A special case which often occurs in practice is when the sources are nearly piecewise stationary. In this case, the pdf $f_{S_i(n)}(.)$ is nearly constant on an interval of τ samples.

3) Each source is stationary but possibly autocorrelated. In this case, we need to model the autocorrelation if we want to simplify the cost function (6). A general and practical solution consists in assuming the sources to be q-th order Markov processes such that

$$f_{S_i|n-1}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(1)) = f_{S_i|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \quad \forall n > q, \tag{9}$$

where $f_{S_i|n-1}(.)$ (respectively $f_{S_i|q}(.)$) denotes the conditional pdf of source s_i given $n-1$ (respectively q) previous samples. The Markov model can take into account the nonlinear correlation between the signal samples. Thus, we can write using the Bayes rule

$$f_{S_i}^{(N)}(\hat{s}_i(1), \ldots, \hat{s}_i(N)) = f_{S_i}^{(q)}(\hat{s}_i(1), \ldots, \hat{s}_i(q)) \times \prod_{n=q+1}^{N} f_{S_i|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)), \tag{10}$$

where the pdf do not depend on n, thanks to stationarity. Consequently, we have

$$C = \left(\frac{-1}{N} \sum_{i=1}^{K} \log f_{S_i}^{(q)}(\hat{s}_i(1), \ldots, \hat{s}_i(q)) \right) + \left(\frac{-1}{N} \sum_{i=1}^{K} \sum_{n=q+1}^{N} \log f_{S_i|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \right) + \frac{1}{N} \sum_{n=1}^{N} \log |\hat{J}(n)|. \tag{11}$$

4) Each source is possibly non-stationary and autocorrelated. Using the Markov model, we can write

$$C = \left(\frac{-1}{N} \sum_{i=1}^{K} \log f_{S_i}^{(q)}(\hat{s}_i(1), \ldots, \hat{s}_i(q)) \right) + \left(\frac{-1}{N} \sum_{i=1}^{K} \sum_{n=q+1}^{N} \log f_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \right) + \frac{1}{N} \sum_{n=1}^{N} \log |\hat{J}(n)|, \tag{12}$$

where $f_{S_i(n)|q}(.)$ denotes the conditional pdf of the source s_i at time n given q previous samples, which depends on n. Once more, a special case is when the sources are piecewise stationary on intervals of length τ.

In the following, we develop our proposed method for the last case because the other cases may be considered as special cases of this one. We suppose that all sources are q-th order Markov processes and piecewise stationary on intervals of length τ.

B. Gradient and Hessian of the Cost Function

Minimizing the cost function (12) typically involves the computation of its gradient with respect to the parameter vector $\hat{\boldsymbol{\theta}}$. Note that the term $f_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))$ in (12) depends on the $q+1$ samples $\hat{s}_i(n), \hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)$ so that its gradient with respect to $\hat{\boldsymbol{\theta}}$ reads

$$\frac{d \log f_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))}{d\hat{\boldsymbol{\theta}}} = \sum_{l=0}^{q} \frac{\partial \log f_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))}{\partial \hat{s}_i(n-l)} \frac{\partial \hat{s}_i(n-l)}{\partial \hat{\boldsymbol{\theta}}}. \tag{13}$$

Let's define the conditional score function of source s_i at time n with respect to the sample $s_i(n-l)$ as follows:

$$\psi_{S_i(n)|q}^{l}(s_i(n)|s_i(n-1), \ldots, s_i(n-q)) = -\frac{\partial \log f_{S_i(n)|q}(s_i(n)|s_i(n-1), \ldots, s_i(n-q))}{\partial s_i(n-l)} \quad \forall l \in [0, q]. \tag{14}$$

We also suppose $N \gg q$ so that the first term in (12), corresponding to the joint pdf of the first q samples of each source, is negligible compared to the second term. We also assume that

the gradient of the first term is negligible compared to the gradient of the second term. Since $\frac{\partial \log|\hat{J}|}{\partial \hat{\boldsymbol{\theta}}} = \frac{1}{\hat{J}}\frac{\partial \hat{J}}{\partial \hat{\boldsymbol{\theta}}}$, the gradient reads

$$\frac{dC}{d\hat{\boldsymbol{\theta}}} \simeq \left(\frac{1}{N}\sum_{i=1}^{K}\sum_{n=q+1}^{N}\sum_{l=0}^{q}\psi^l_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))\frac{\partial \hat{s}_i(n-l)}{\partial \hat{\boldsymbol{\theta}}}\right) + \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\hat{J}(n)}\frac{\partial \hat{J}(n)}{\partial \hat{\boldsymbol{\theta}}}. \quad (15)$$

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

$$\mathbf{0} = \frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\boldsymbol{\theta}}}\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\mathbf{s}}(n)} + \frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}(n)\ cst}, \quad (16)$$

which implies

$$\frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\boldsymbol{\theta}}} = -\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}(n)\ cst}\left(\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\mathbf{s}}(n)}\right)^{-1}, \quad (17)$$

where "$\hat{\mathbf{s}}\ cst$" stands for "$\hat{\mathbf{s}}$ is constant". In the above expressions, $\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\mathbf{s}}(n)}$ is the $K \times K$ Jacobian matrix of the mixing model, with the generic entry $\left(\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\mathbf{s}}(n)}\right)_{i,j} = \frac{\partial \mathcal{F}_j(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{s}_i(n)}$ and the entry (i,j) of the matrix $\frac{\partial \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\boldsymbol{\theta}}}_{|\hat{\mathbf{s}}(n)\ cst}$ (respectively $\frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\boldsymbol{\theta}}}$) is $\frac{\partial \mathcal{F}_j(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})}{\partial \hat{\theta}_i}_{|\hat{\mathbf{s}}(\mathbf{n})\ cst}$ (respectively $\frac{\partial \hat{s}_j(n)}{\partial \hat{\theta}_i}$). Inserting the entries of (17) evaluated at times $n, n-1, \ldots, n-q$ in (15), we get the expression of the gradient which may be used for minimizing the cost function using e.g., a gradient descent algorithm

$$\hat{\boldsymbol{\theta}}_{new} = \hat{\boldsymbol{\theta}}_{old} - \mu\frac{dC}{d\hat{\boldsymbol{\theta}}}, \quad (18)$$

where μ is a positive learning rate[4]. This learning rate may be chosen using a line search algorithm satisfying the Armijo condition. In our simulations, we used a small constant learning rate ensuring convergence (the convergence was visually checked).

The above approach based on implicit differentiation does not require the knowledge of the explicit inverse of the mixing model. According to (15) and (17), we only need to know the parametric expression of the mixing model $\boldsymbol{\mathcal{F}}$ (and not its inverse) in order to obtain the *analytical expression* of the gradient $\frac{dC}{d\hat{\boldsymbol{\theta}}}$, up to the approximation made in (15).

Nevertheless, to calculate the *numerical value* of the gradient from expression (15), 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. These samples may be computed by solving the equation $\mathbf{x}(n) = \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})$ at each time n using e.g., a numerical algorithm.

To avoid the convergence issues related to the choice of learning rate in the gradient descent method, we may prefer to use Newton's algorithm. From (15), the entry (k,m) of the Hessian matrix $\mathbf{H}$, up to the approximation made in (15), can be computed as follows:

$$\begin{aligned} H_{km} &= \frac{d}{d\theta_m}\frac{dC}{d\theta_k} \simeq \left(\frac{1}{N}\sum_{i=1}^{K}\sum_{n=q+1}^{N}\sum_{l=0}^{q}\right. \\ &\left(\psi^l_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))\frac{\partial}{\partial \hat{\theta}_m}\frac{\partial \hat{s}_i(n-l)}{\partial \hat{\theta}_k}\right. \\ &+ \sum_{j=0}^{q}\frac{\partial \psi^l_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q))}{\partial \hat{s}_i(n-j)} \\ &\left.\left.\times \frac{\partial \hat{s}_i(n-j)}{\partial \hat{\theta}_m}\frac{\partial \hat{s}_i(n-l)}{\partial \hat{\theta}_k}\right)\right) + \frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial \hat{\theta}_m}\left(\frac{1}{\hat{J}(n)}\frac{\partial \hat{J}(n)}{\partial \hat{\theta}_k}\right). \end{aligned} \quad (19)$$

Then, the Newton's update rule reads

$$\hat{\boldsymbol{\theta}}_{new} = \hat{\boldsymbol{\theta}}_{old} - \mathbf{H}^{-1}\frac{dC}{d\hat{\boldsymbol{\theta}}}, \quad (20)$$

assuming the invertibility of $\mathbf{H}$. If $\mathbf{H}$ is close to a non-invertible matrix, we can modify it e.g., by adding a correction matrix $\mathbf{D}$ so as to make $\mathbf{H}+\mathbf{D}$ positive-definite. A common choice of $\mathbf{D}$, used in the Levenberg-Marquardt algorithm, is a scaled identity matrix.

C. Score Function Estimation

In practice, the conditional pdf $f_{S_i(n)|q}(.)$ and the conditional score functions $\psi^l_{S_i(n)|q}(.)$ of the actual sources are usually unknown. Like in linear BSS [4], [30] we can replace them by the estimated score functions of the signals $\hat{s}_i(n)$, determined as mentioned above, in each iteration of the gradient algorithm.

These score functions may be for example estimated using the approach proposed in [4] (and inspired from [30] and [36]). In this approach, each conditional score function is first written as the difference of two joint score functions as follows:

$$\begin{aligned} &\psi^l_{S_i(n)|q}(\hat{s}_i(n)|\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \\ &\quad = \psi^l_{S_i(n),q+1}(\hat{s}_i(n), \hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \\ &\quad\quad - \psi^l_{S_i(n),q}(\hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)). \end{aligned} \quad (21)$$

The first joint score function in (21) may be estimated by writing

$$\begin{aligned} &\psi^l_{S_i(n),q+1}(\hat{s}_i(n), \hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)) \\ &\quad = \sum_{m=1}^{M} c^l_m \phi_m(\hat{s}_i(n), \hat{s}_i(n-1), \ldots, \hat{s}_i(n-q)), \end{aligned} \quad (22)$$

where $\phi_m(.)$ are some basis functions, and by computing the coefficients c^l_m which are the solutions of the following equation (see [4] and [30] for details):

$$\mathbf{G}[c^l_1, \ldots, c^l_M]^T = \mathbf{g}, \quad (23)$$

where $\mathbf{G} = E[\boldsymbol{\phi}\boldsymbol{\phi}^T]$, $\mathbf{g} = E[\boldsymbol{\phi}']$ with E denoting the expectation operator, $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_M]^T$, and $\boldsymbol{\phi}'$ its derivative with respect to $\hat{s}_i(n-l)$. These derivatives may also be used for

[4]We may also use a momentum term to improve the convergence of the gradient algorithm.

estimating the score function derivatives required in Newton's method.

As can be seen, the estimation of the coefficients c_m^l requires the computation of some expected values. Nevertheless, this is not possible unless we make some statistical assumptions about the sources. If the sources are stationary (and ergodic), the expected values may be replaced by sample means. If the sources are non-stationary but piecewise stationary, we can use the blocking method mentioned in [4] (and inspired from [37]). In this method, the piecewise stationary signal, of length N, is split into L adjacent intervals I_j, $j = 1, \ldots, L$, each one containing $\tau = N/L$ samples. Under the piecewise stationarity (i.e., slow variation) hypothesis, score functions are supposed to be constant within each of the intervals I_j. Thus, in each I_j, the score functions do not depend on time and the expected values may be replaced by sample means on the intervals.

The second joint score function in (21) may be estimated in the same manner using other basis functions and coefficients, thus providing, using (21), an estimate of conditional score functions on each interval.

D. Overall Algorithm

In summary, we propose the following algorithm for blindly separating a parametric nonlinear mixture of possibly non-stationary and autocorrelated sources:

1) Initialize the mixing parameter estimates $\hat{\boldsymbol{\theta}}$ (with random values if there is no information about them).
2) Repeat the following steps until convergence (i.e., until the mixing parameter estimates $\hat{\boldsymbol{\theta}}$ do not significantly change):
   - Use the current estimate of the mixing parameters for computing a new estimate of the sources, $\hat{\mathbf{s}}(n)$, for each $n \in [1, N]$ by solving (analytically or numerically) the mixing equation $\mathbf{x}(n) = \boldsymbol{\mathcal{F}}(\hat{\mathbf{s}}(n), \hat{\boldsymbol{\theta}})$.
   - Estimate the conditional score functions of the estimated sources on L intervals I_j of length τ.[5]
   - Compute the gradient using (15) and (17), and possibly the Hessian using (19).
   - Update the mixing parameter estimates $\hat{\boldsymbol{\theta}}$ using an optimization algorithm like gradient descent or Newton's algorithm.

Note that if the signals are stationary, we can apply the above algorithm by choosing $L = 1$ (i.e., $\tau = N$) and if each signal has independent samples, we can apply the algorithm by choosing $q = 0$ (i.e., a zero-order Markov model).

The main hypotheses under which our method can be applied are listed below.

1) The sources must be independent.
2) The parametric model of the memoryless mixing transformation must be known except for some unknown parameter values.
3) The mixing model must be bijective in the variation domain of the sources.
4) The mixing model must be differentiable with respect to the sources and the mixing parameters.

[5]The optimal length of these intervals depends on the nature of the sources. For example, it is well known that speech signals are relatively stationary on short intervals of 10–20 milliseconds.

In the following two sections, we theoretically illustrate this proposed algorithm for a bijective toy mixture and a general linear-quadratic mixture, and we present some simulation results using the toy model and using a subclass of the linear-quadratic model called the bilinear model.

III. A Simple Bijective "Toy" Model

The first example studied in this paper is a simple "toy" mixing model with known inverse which is globally bijective. While this model does not fit any known physical system, its study will be useful because of its simplicity and its global bijectivity which allows us to focus our efforts on parameter estimation.

The 2×2 mixing model is defined by a single parameter θ:

$$\begin{aligned} x_1(n) &= s_1(n)^3 + \theta s_2(n) \\ x_2(n) &= -\theta s_1(n) + s_2(n). \end{aligned} \tag{24}$$

In the following, we compute its inverse, omitting the time index n for simplifying the notation. The mixing (24) yield

$$s_1^3 + \theta^2 s_1 + \theta x_2 - x_1 = 0, \tag{25}$$

which can be solved using Cardano's formula with respect to s_1 to obtain one of the separating equations. Let's denote $r = \theta x_2 - x_1$, $p = \theta^2$ and $\Delta = \frac{r^2}{4} + \frac{p^3}{27}$. If $\theta \neq 0$, then $\Delta > 0$ so that the cubic (25) has a unique real root defined by

$$s_1 = \left(\frac{-r}{2} + \sqrt{\Delta}\right)^{1/3} + \left(\frac{-r}{2} - \sqrt{\Delta}\right)^{1/3}. \tag{26}$$

The other source may then be obtained using

$$s_2 = x_2 + \theta s_1. \tag{27}$$

If $\theta = 0$, then the mixing (24) has a unique real solution $(s_1, s_2) = (x_1^{1/3}, x_2)$. Thus, the mixture is globally bijective everywhere in $\mathbb{R}^2$.

In the following, we successively study the case of i.i.d. sources and that of non-i.i.d. sources, then present some simulation results.

A. Case of I.I.D. Sources

The Jacobian of the mixing (24) reads

$$J(n) = 3s_1(n)^2 + \theta^2. \tag{28}$$

Replacing in (7), we obtain the cost function to be minimized. From (15), the gradient of this cost function is equal to (see Appendix A for the computational details):

$$\begin{aligned} \frac{\partial C}{\partial \hat{\theta}} = \frac{1}{N} \sum_{n=1}^{N} \Bigg(& \psi_{S_1}^0(\hat{s}_1(n)) \frac{-(\hat{s}_2(n) + \hat{\theta}\hat{s}_1(n))}{3\hat{s}_1(n)^2 + \hat{\theta}^2} \\ & + \psi_{S_2}^0(\hat{s}_2(n)) \frac{-(\hat{\theta}\hat{s}_2(n) - 3\hat{s}_1(n)^3)}{3\hat{s}_1(n)^2 + \hat{\theta}^2} \\ & + \frac{1}{3\hat{s}_1(n)^2 + \hat{\theta}^2} \left[6\hat{s}_1(n) \frac{-(\hat{s}_2(n) + \hat{\theta}\hat{s}_1(n))}{3\hat{s}_1(n)^2 + \hat{\theta}^2} + 2\hat{\theta} \right] \Bigg). \end{aligned} \tag{29}$$

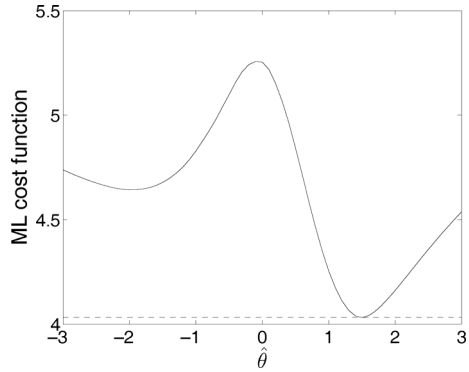

Fig. 1. Cost function using known pdf.

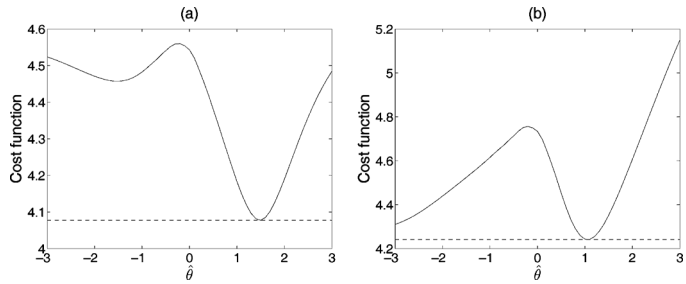

Fig. 2. Cost function using unknown pdf: (a) with pdf shape re-estimated for each value of $\hat{\theta}$, (b) with pdf shape estimated for $\hat{\theta} = 1$.

B. Case of Non-I.I.D. Sources

When sources are not i.i.d., the gradient (15) of the general cost function (12) can be computed in the same manner as above but by choosing $q \neq 0$ in (15) and score functions possibly depending on n. For example, in the case of non-stationary 1st-order Markov sources ($q = 1$), this gradient reads:

$$\begin{aligned}
&\frac{\partial C}{\partial \hat{\theta}} \\
&= \frac{1}{N}\sum_{n=2}^{N}\left(\psi^0_{S_1(n)|1}(\hat{s}_1(n)|\hat{s}_1(n-1))\frac{-(\hat{s}_2(n)+\hat{\theta}\hat{s}_1(n))}{3\hat{s}_1(n)^2+\hat{\theta}^2}\right. \\
&\quad + \psi^1_{S_1(n)|1}(\hat{s}_1(n)|\hat{s}_1(n-1))\frac{-(\hat{s}_2(n-1)+\hat{\theta}\hat{s}_1(n-1))}{3\hat{s}_1(n-1)^2+\hat{\theta}^2} \\
&\quad + \psi^0_{S_2(n)|1}(\hat{s}_2(n)|\hat{s}_2(n-1))\frac{-(\hat{\theta}\hat{s}_2(n)-3\hat{s}_1(n)^3)}{3\hat{s}_1(n)^2+\hat{\theta}^2} \\
&\quad \left. + \psi^1_{S_2(n)|1}(\hat{s}_2(n)|\hat{s}_2(n-1))\frac{-(\hat{\theta}\hat{s}_2(n-1)-3\hat{s}_1(n-1)^3)}{3\hat{s}_1(n-1)^2+\hat{\theta}^2}\right) \\
&\quad + \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{3\hat{s}_1(n)^2+\hat{\theta}^2}\left[6\hat{s}_1(n)\frac{-(\hat{s}_2(n)+\hat{\theta}\hat{s}_1(n))}{3\hat{s}_1(n)^2+\hat{\theta}^2}+2\hat{\theta}\right]\right).
\end{aligned} \tag{30}$$

C. Simulation Results

1) Separating I.I.D. Sources Supposing Known Source PDF: In the first experiment, we generated two unit-variance i.i.d and mutually independent Laplacian sources of $N = 1000$ samples, then mixed them using the mixing model (24) with $\theta = 1.5$. Fig. 1 shows the cost function (7) using the actual value of $f_{S_i}(\hat{s}_i(n))$, i.e., $\frac{\sqrt{2}}{2}\exp(-\sqrt{2}|\hat{s}_i(n)|)$ for $\hat{\theta} \in [-3, 3]$ where $\hat{s}_i(n)$ and $\hat{J}(n)$ are computed from (26), (27) and (28) replacing θ by $\hat{\theta}$. As can be seen, this function has two local minima but only one of them (which is also the global minimum) corresponds to the actual value of the parameter and provides the independent components corresponding to the actual sources. When initialized with negative values of $\hat{\theta}$, the optimization algorithms like gradient descent or Newton converge towards this spurious local minimum. However, this value may be rejected a posteriori using an independence test.

2) Separating I.I.D. Sources Supposing Unknown Source PDF: The above experiment was repeated supposing that the source pdf are unknown. Thus, the actual score functions were replaced by the estimated score functions of the signals $\hat{s}_i(n)$ using the method described in the previous section. In this case, maximizing the likelihood is asymptotically equivalent to minimizing the mutual information between the output signals $\hat{s}_i(n)$.

The cost function (7), but using $f_{\hat{s}_i}(.)$ instead of $f_{s_i}(.)$, is shown in Fig. 2(a) as a function of $\hat{\theta}$. Once more, the cost function has a global minimum corresponding to the actual parameter value and another, spurious, local minimum which may be rejected using an independence test. Fig. 3 shows the scatter plots of the mixtures and of the output components corresponding to these minima. Note also that in practice, at each iteration of an optimization algorithm, one first estimates (using the current value of the parameter $\hat{\theta}$) the coefficients c^l_m in (23) which determine the shape of score functions (and related pdf), then freezes them and performs a minimization step for the cost function related to these pdf with respect to $\hat{\theta}$. Since the estimated pdf change during successive iterations, the shape of the function to be minimized changes too. For example, Fig. 2(b) shows the cost function as a function of $\hat{\theta}$ in the above example (i.e., with $\theta = 1.5$) corresponding to the coefficients c^l_m estimated using the value $\hat{\theta} = 1$. As can be seen, this function is not the same as in Fig. 2(a). The practical optimization is therefore more difficult than what may be suggested by Fig. 2(a) because the shape of the function to be minimized changes at each iteration of the gradient algorithm. This example also shows the sensitivity of the method to the estimation of score functions: if the estimated score functions are not updated in the following iterations, the optimization algorithm converges towards the minimum of Fig. 2(b), i.e., $\hat{\theta} = 1.06$.

3) Simulations Using Autocorrelated Sources: In this experiment, we want to show that when each source is autocorrelated, better performance can be obtained by taking into account this autocorrelation using a Markov model.

We generated two mutually independent i.i.d. signals $e_1(n)$ and $e_2(n)$ uniformly distributed over $[-0.5, 0.5]$, that we filtered by two autoregressive filters in order to obtain two stationary 1st-order Markov sources following the scheme $s_i(n) = e_i(n) + \rho_i s_i(n-1)$. The chosen coefficients were $\rho_1 = 0.7$ and $\rho_2 = 0.9$. The sources were then normalized to have unit variances. The mixture was generated using the model (24) with $\theta = 1.5$.

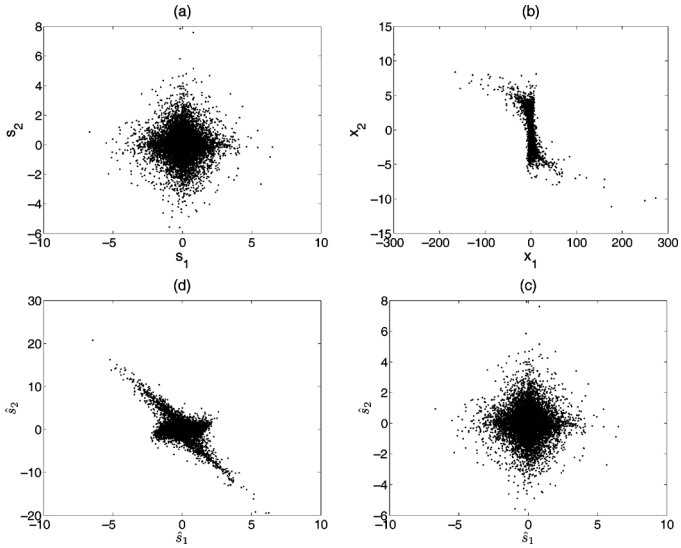

Fig. 3. scatter plots of (a) sources, (b) mixtures, (c) output components corresponding to the desired local minimum, (d) output components corresponding to the spurious local minimum.

TABLE I
MEAN AND STANDARD DEVIATION OF SIR, AND MEAN OF COMPUTATION TIME, WITHOUT AND WITH TAKING INTO ACCOUNT THE COLOR OF SOURCES

Method	mean(SIR)	std(SIR)	mean(time)
i.i.d.-based gradient	34.6 dB	9.8 dB	0.27 sec
Markov-based gradient	51.0 dB	11.2 dB	0.04 sec

We used the algorithm proposed in the previous section for separating the sources using gradient descent, first ignoring the time structure (i.e., using the gradient (29) to minimize the cost function (7) which is optimal, in the ML sense, only for i.i.d. sources), then taking into account this structure using the gradient (30). In each case, we performed 100 Monte Carlo simulations corresponding to 100 different initial values of the random source generator and of the parameter $\hat{\theta}$. For each simulation, the Signal to Interference Ratio (SIR) was computed by

$$SIR = \frac{1}{K}\sum_{i=1}^{K} 10\log_{10}\frac{\sum_{n=1}^{N} s_i(n)^2}{\sum_{n=1}^{N}(\tilde{s}_i(n)-s_i(n))^2}, \qquad (31)$$

where $\tilde{s}_i(n)$ is the final estimate of $s_i(n)$ after normalizing it so that it has the same variance and sign as $s_i(n)$, and $K = 2$. The mean and the standard deviation of SIR over 100 Monte Carlo simulations for $N = 1000$ source samples are shown in Table I. As can be seen, performance is better when taking autocorrelation into account. This table also shows the average running time of our methods using a non-optimized Matlab code on a computer with an Intel core 2 Quad CPU, with a frequency of 2.8 GHz and a RAM of 4 GB.

4) Simulations Using Non-Stationary Sources: In this experiment, we want to highlight the relevance of taking into account the possible non-stationarity of the signals in nonlinear BSS.

First, we generated two mutually independent i.i.d. signals $e_1(n)$ and $e_2(n)$ uniformly distributed over $[-0.5, 0.5]$. Then, we split these signals into two intervals and multiplied the first interval by 7 to obtain the non-stationary source signals $s_i(n)$. Finally, the sources were normalized to have unit variances and mixed using the model (24) with $\theta = 1.5$. The first source is shown in Fig. 4. We used the algorithm proposed in the previous section for separating the sources using gradient descent, first ignoring the non-stationarity (i.e., using the gradient (29) which is optimal, in the ML sense, only for i.i.d. sources), then taking into account this non-stationarity. In the first case, the score functions were estimated on the whole signals while in the second case, they were estimated separately on each interval using the blocking method described in the previous section.

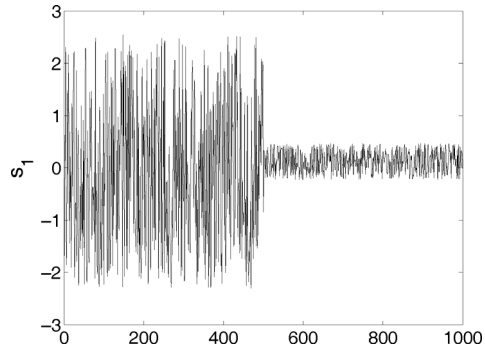

Fig. 4. One of the non-stationary sources used in the simulations.

TABLE II
MEAN AND STANDARD DEVIATION OF SIR, AND MEAN OF COMPUTATION TIME, WITHOUT AND WITH TAKING INTO ACCOUNT THE NON-STATIONARITY OF SOURCES

Method	mean(SIR)	std(SIR)	mean(time)
i.i.d.-based gradient	7.4 dB	0.5 dB	0.08 sec
non-stationary-based gradient	42.7 dB	8.7 dB	0.05 sec

TABLE III
MEAN OF SIR AS A FUNCTION OF SAMPLE SIZE N WITH TAKING INTO ACCOUNT THE NON-STATIONARITY OF SOURCES

N	50	100	500	1000
mean(SIR) in dB	24.4	31.5	39.4	42.7

In each case, we performed 100 Monte Carlo simulations corresponding to 100 different initial values of the random source generator and of the parameter $\hat{\theta}$. The mean and the standard deviation of SIR, as well as the average running time, over 100 simulations for $N = 1000$ source samples are shown in Table II. As can be seen, performance is very poor when non-stationarity is ignored, and very satisfactory when it is taken into account. Finally, Table III shows the mean of SIR as a function of the sample size N in the latter case.

IV. LINEAR-QUADRATIC MODEL

In this section, we study the linear-quadratic mixing model, which can be represented in the general case by

$$x_i(n) = \sum_{k=1}^{K}\rho_{ik}u_k(n) + \sum_{k=1}^{K}\gamma_{ik}u_k^2(n) + \sum_{k=1}^{K-1}\sum_{l=k+1}^{K}\xi_{i(k,l)}u_k(n)u_l(n) \qquad i = 1,\ldots,K. \qquad (32)$$

In the context of ICA, such models have been studied in the case of circular complex sources and strictly over-determined mixtures (i.e., more observations than sources) [16], [17], binary sources and strictly over-determined mixtures [20], two mixtures of two i.i.d. sources (without square terms of the sources) [18], [35], [38]. In this section, however, we study this model in a general determined configuration (same arbitrary number of observations and sources) and we suppose that the sources are arbitrary real signals, possibly autocorrelated and/or non-stationary. We aim at using the approach presented in Section II to achieve BSS and to measure the influence of taking into account the autocorrelation and/or non-stationarity of sources on performance. The linear-quadratic model has recently been used to describe physical mixing phenomena like the show-through effect in scanned documents which yields mixtures of the front and back images of a thin paper [21], [39], or the multiple scattering of light between surfaces which yields nonlinearly mixed pixels in multi/hyperspectral remote sensing of non-flat surfaces [40]–[43]. In the latter case, methods based on sparsity [44] or non-negativity [45] of data have been proposed to unmix them.

Supposing the signals $u_k(n)$ are independent, we want to estimate them up to a permutation and a scaling factor. In order to define a simple convention to handle the scale indeterminacy of ICA, we hereafter consider the following normalized equivalent model

$$x_i(n) = s_i(n) + \sum_{k=1,k\neq i}^{K} a_{ik}s_k(n) + \sum_{k=1}^{K} b_{ik}s_k^2(n) + \sum_{k=1}^{K-1}\sum_{l=k+1}^{K} d_{i(k,l)}s_k(n)s_l(n), \quad (33)$$

where $s_i(n) = \rho_{ii}u_i(n)$, $a_{ik} = \rho_{ik}/\rho_{kk}$, $b_{ik} = \gamma_{ik}/\rho_{kk}^2$, $d_{i(k,l)} = \xi_{i(k,l)}/(\rho_{kk}\rho_{ll})$. The model is defined by an unknown parameter vector $\boldsymbol{\theta}$ which contains all the coefficients a_{ik}, b_{ik} and $d_{i(k,l)}$ in (33).

A. Computing the Gradient

Using the mixing model (33), we can compute the entries of the Jacobian matrix $\boldsymbol{\mathcal{J}}(n) = \frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \mathbf{s}(n)}$ as follows:

$$\mathcal{J}_{ji}(n) = \frac{\partial x_i(n)}{\partial s_j(n)} = a_{ij} + 2b_{ij}s_j(n) + \sum_{k=1,k\neq j}^{K} d_{i(k,j)}s_k(n), \quad (34)$$

where we use the following notation conventions: $a_{ii} = 1$ and $d_{i(k,j)} = d_{i(j,k)}$. Computing the derivatives of (33) with respect to the parameters, keeping $\mathbf{s}(n)$ constant, yields

$$\frac{\partial x_i(n)}{\partial a_{lj}}_{|_{\mathbf{s}(n)\ cst}} = \begin{cases} s_j(n) & l = i \neq j \\ 0 & \text{otherwise} \end{cases}, \quad (35)$$

$$\frac{\partial x_i(n)}{\partial b_{lj}}_{|_{\mathbf{s}(n)\ cst}} = \begin{cases} s_j^2(n) & l = i \\ 0 & \text{otherwise} \end{cases}, \quad (36)$$

$$\frac{\partial x_i(n)}{\partial d_{l(m,j)}}_{|_{\mathbf{s}(n)\ cst}} = \begin{cases} s_m(n)s_j(n) & l = i, j > m \\ 0 & \text{otherwise.} \end{cases} \quad (37)$$

The above three equations thus allow one to compute $\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}(n)\ cst}}$. Replacing $\mathbf{s}$ and $\boldsymbol{\theta}$ by $\hat{\boldsymbol{s}}$ and $\hat{\boldsymbol{\theta}}$ in the above expressions of $\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \mathbf{s}(n)}$ and $\frac{\partial \boldsymbol{\mathcal{F}}(\mathbf{s}(n),\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}(n)\ cst}}$, then using (17), 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 (15). To determine the second term, we need to compute $\frac{\partial J(n)}{\partial \boldsymbol{\theta}}$. Denoting $J(n) = g(\boldsymbol{\theta}, \mathbf{s}(n))$ and considering $\boldsymbol{\theta}$ as the independent variable and $\mathbf{s}(n)$ as a function of this variable, we can write

$$\frac{\partial J(n)}{\partial \boldsymbol{\theta}} = \frac{\partial J(n)}{\partial \boldsymbol{\theta}}_{|_{\mathbf{s}(n)\ cst}} + \frac{\partial \mathbf{s}(n)}{\partial \boldsymbol{\theta}}\frac{\partial J(n)}{\partial \mathbf{s}(n)}. \quad (38)$$

The entries of the first term in the above equation may be computed as follows with $\mathtt{tr}\{\boldsymbol{\mathcal{J}}(\boldsymbol{n})\}$, $\mathtt{adj}(\boldsymbol{\mathcal{J}}(n))$, and $\mathtt{cof}(\boldsymbol{\mathcal{J}}(n))$ respectively standing for trace, adjoint matrix, and matrix of cofactors of $\boldsymbol{\mathcal{J}}(n)$ [46]:

$$\begin{aligned} \frac{\partial J(n)}{\partial \theta_i}_{|_{\mathbf{s}(n)\ cst}} &= J(n)\cdot \mathtt{tr}\left\{ \boldsymbol{\mathcal{J}}(n)^{-1}\frac{\partial \boldsymbol{\mathcal{J}}(n)}{\partial \theta_i}_{|_{\mathbf{s}(n)\ cst}} \right\} \\ &= J(n)\cdot \mathtt{tr}\left\{ \frac{1}{J(n)}\mathtt{adj}(\boldsymbol{\mathcal{J}}(n))\frac{\partial \boldsymbol{\mathcal{J}}}{\partial \theta_i}_{|_{\mathbf{s}(n)\ cst}} \right\} \\ &= \mathtt{tr}\left\{ (\mathtt{cof}(\boldsymbol{\mathcal{J}}(n)))^T\frac{\partial \boldsymbol{\mathcal{J}}(n)}{\partial \theta_i}_{|_{\mathbf{s}(n)\ cst}} \right\}. \end{aligned} \quad (39)$$

For example, to compute the derivative with respect to the parameter a_{12}, we know from (34) that

$$\frac{\partial \mathcal{J}_{ji}(n)}{\partial a_{12}}_{|_{\mathbf{s}(n)\ cst}} = \begin{cases} 0 & \forall (j,i) \neq (2,1) \\ 1 & (j,i) = (2,1) \end{cases}, \quad (40)$$

which yields, denoting by $\mathtt{cof}(\boldsymbol{\mathcal{J}}(n), i, j)$ the (i,j) cofactor of matrix $\boldsymbol{\mathcal{J}}(n)$:

$$\begin{aligned} &\frac{\partial J(n)}{\partial a_{12}}_{|_{\mathbf{s}(n)\ cst}} \\ &= \mathtt{tr}\left\{ \begin{pmatrix} \mathtt{cof}(\boldsymbol{\mathcal{J}}(n),1,1) & \cdots & \mathtt{cof}(\boldsymbol{\mathcal{J}}(n),K,1) \\ \vdots & \ddots & \vdots \\ \mathtt{cof}(\boldsymbol{\mathcal{J}}(n),1,K) & \cdots & \mathtt{cof}(\boldsymbol{\mathcal{J}}(n),K,K) \end{pmatrix} \right. \\ &\quad \left. \times \begin{pmatrix} 0 & 0 & \cdots \\ 1 & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \ddots & \vdots \\ 0 & 0 & \cdots \end{pmatrix} \right\} \\ &= \mathtt{cof}(\boldsymbol{\mathcal{J}}(n),2,1). \end{aligned} \quad (41)$$

The entries of $\frac{\partial J(n)}{\partial \mathbf{s}}$ may be computed using

$$\begin{aligned} \frac{\partial J(n)}{\partial s_l(n)} &= J(n)\cdot \mathtt{tr}\left\{ \boldsymbol{\mathcal{J}}(n)^{-1}\frac{\partial \boldsymbol{\mathcal{J}}(n)}{\partial s_l(n)} \right\} \\ &= \mathtt{tr}\left\{ (\mathtt{cof}(\boldsymbol{\mathcal{J}}(n)))^T\frac{\partial \boldsymbol{\mathcal{J}}(n)}{\partial s_l(n)} \right\}, \end{aligned} \quad (42)$$

and noting that, from (34)

$$\frac{\partial \mathcal{J}_{ji}(n)}{\partial s_l(n)} = \begin{cases} 2b_{il} & j = l \\ d_{i(l,j)} & j \neq l \end{cases}. \quad (43)$$

Replacing J, s and θ by $\hat{J}$, $\hat{s}$ and $\hat{\theta}$ in the expressions of $\frac{\partial J(n)}{\partial \boldsymbol{\theta}}|_{\mathbf{s}(n)\ cst}$ and $\frac{\partial J(n)}{\partial \mathbf{s}(n)}$, 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 (38), then the second term of (15), and therefore the entire gradient $\frac{dC}{d\hat{\boldsymbol{\theta}}}$.

B. Simulation Results

1) Separating I.I.D. Sources: In a first experiment, we consider a special case of the normalized mixing model (33) with $K = 3$ sources and observations where the coefficients of the squared terms, b_{ik}, are set to zero. This special model is usually called the *bilinear* model. The other coefficients are

$$\begin{aligned}\boldsymbol{\theta} &= \{a_{12}, a_{13}, a_{21}, a_{23}, a_{31}, a_{32}, d_{1(1,2)}, d_{1(1,3)}, d_{1(2,3)},\\ &\quad d_{2(1,2)}, d_{2(1,3)}, d_{2(2,3)}, d_{3(1,2)}, d_{3(1,3)}, d_{3(2,3)}\}\\ &= \{1, -1, -1, -1, 1, 1, 0.8, 0.7,\\ &\quad 0.6, 0.5, 0.8, 0.7, 0.8, 0.4, 0.3\}. \end{aligned} \tag{44}$$

Thus, the mixing model is defined by

$$\begin{aligned} x_1(n) &= s_1(n) + s_2(n) - s_3(n) + 0.8 s_1(n) s_2(n)\\ &\quad + 0.7 s_1(n) s_3(n) + 0.6 s_2(n) s_3(n)\\ x_2(n) &= -s_1(n) + s_2(n) - s_3(n) + 0.5 s_1(n) s_2(n)\\ &\quad + 0.8 s_1(n) s_3(n) + 0.7 s_2(n) s_3(n)\\ x_3(n) &= s_1(n) + s_2(n) + s_3(n) + 0.8 s_1(n) s_2(n)\\ &\quad + 0.4 s_1(n) s_3(n) + 0.3 s_2(n) s_3(n), \end{aligned} \tag{45}$$

where $s_1(n)$, $s_2(n)$ and $s_3(n)$ are three 1000-sample i.i.d. sources, uniformly distributed over $[-0.5, 0.5]$. As shown in Appendix B, this choice of parameter values guarantees the global univalence of the mixture in $[-0.5, 0.5]$. We use the following algorithm for estimating the mixing parameters and separating the sources:

– Initialize $\hat{\boldsymbol{\theta}}$ to an initial value $\hat{\boldsymbol{\theta}}_0$.

repeat

– Compute all $\hat{s}_i(n)\ ,\ n = 1, \ldots, N$ using the observations $x_i(n)$ and the current values of the estimated mixing parameters $\hat{\boldsymbol{\theta}}$.

– Estimate the score functions of $\hat{s}_i$.

– Compute the gradient $\frac{\partial C}{\partial \hat{\boldsymbol{\theta}}}$ using the formulas presented in the previous subsection.

– Update the estimated parameter vector using **(18)**.

until convergence (i.e., until the estimated parameters do not significantly change)

– Set the estimated sources $\tilde{s}_i$ to $\hat{s}_i$.

When using an unconstrained optimization algorithm, the estimated mixing parameters may take unacceptable values during the optimization procedure, leading e.g., to non-invertible models. In our tests, we could avoid this issue by initializing the first 6 estimated mixing parameters $\hat{\theta}_1, \ldots, \hat{\theta}_6$ (corresponding to the linear contributions of the sources in the mixture) with their estimates obtained by using a hypothesized linear model which provides a rough approximation of the actual values[6]. In this approach, we first suppose that the mixtures are generated using the following linear model

$$\begin{aligned} x_1(n) &= s_1(n) + \theta_1 s_2(n) + \theta_2 s_3(n)\\ x_2(n) &= \theta_3 s_1(n) + s_2(n) + \theta_4 s_3(n)\\ x_3(n) &= \theta_5 s_1(n) + \theta_6 s_2(n) + s_3(n). \end{aligned} \tag{46}$$

Then, we use the ML linear BSS algorithm of [30] to obtain a first estimate of the mixing parameters which will be used as the initial values of $\hat{\theta}_1, \ldots, \hat{\theta}_6$ in our algorithm. The other parameters $\hat{\theta}_7, \ldots, \hat{\theta}_{15}$ are initialized to small random values in $[-0.05, 0.05]$. To compute $\hat{s}_i(n)$ from the current values of the estimated mixing parameters $\hat{\boldsymbol{\theta}}$, we have to solve a system of three nonlinear equations i.e., the system (45) where s_i are replaced by $\hat{s}_i$ and the actual parameters (θ_i) by their estimates $\hat{\theta}_i$. By eliminating $\hat{s}_1$ and $\hat{s}_2$, we can obtain a unique equation which only depends on $\hat{s}_3$, the observations x_1, x_2, x_3 and the mixing parameters $\hat{\theta}_i$. It can be checked that this equation includes the terms containing $\hat{s}_3^5$, $\hat{s}_3^4$, $\hat{s}_3^3$, $\hat{s}_3^2$, $\hat{s}_3$ and the square roots of some polynomial functions of $\hat{s}_3$. Such an equation cannot analytically be solved for $\hat{s}_3$. Nevertheless, the system of (45) may be solved numerically. In all the simulations described in this section, we used a Gauss-Newton algorithm for this purpose.

The score functions were estimated using the approach described in Section II-C with the basis functions $\phi_m(\hat{s}_i(n)) = \hat{s}_i(n)^{m-1}\ ,\ m = 1, \ldots, 5$. Performance was measured using the SIR defined in (31) with $K = 3$. The algorithm converged after about 500 iterations and led to a 29.3-dB SIR. The same experiment, neglecting the quadratic part in the source separation procedure (i.e., supposing the linear mixing model (46) and using the ML linear BSS algorithm of [30]) led to a 16.3-dB SIR. This result confirms the advantage of taking into account the entire model.

2) Separating Non-Stationary and Autocorrelated Sources: In another experiment, we first generated three mutually independent i.i.d. signals $e_1(n)$, $e_2(n)$ and $e_3(n)$ uniformly distributed over $[-0.5, 0.5]$, that we filtered by three autoregressive filters in order to obtain three stationary 1st-order Markov signals z_i following the scheme $z_i(n) = e_i(n) + \rho_i z_i(n-1)$. The chosen coefficients were $\rho_1 = 0.7$, $\rho_2 = 0.8$ and $\rho_3 = 0.9$. Then, we split each of these signals into two intervals and multiplied the first interval by 2 to obtain non-stationary signals. Finally, the three signals were normalized to be in $[-0.5, 0.5]$. The resulting non-stationary and autocorrelated sources $s_i(n)$ were then mixed using the same mixing model (45) as in the previous simulation which guarantees the global univalence of the mapping in the variation domain of the sources. We used our algorithm for separating the sources, first ignoring the non-stationarity and the autocorrelation of signals, then taking these properties into account. In the first case, we chose $q = 0$ when computing the gradient using (15) and we estimated the *marginal* score functions on the whole signals. In the second case, we chose $q = 1$ (corresponding to a first-order Markov model) and

[6]Another solution is to use a constrained optimization algorithm which keeps the estimated parameters in their acceptable range of variation.

TABLE IV
SIR RESULTS IN THE SECOND SIMULATION WITH BILINEAR MODEL: (A) IGNORING THE NON-STATIONARITY AND THE AUTOCORRELATION OF SIGNALS, (B) TAKING THESE PROPERTIES INTO ACCOUNT

	mean(SIR)	std(SIR)
(A)	8.40 dB	3.43 dB
(B)	30.69 dB	2.07 dB

estimated the $conditional$ score functions separately on each interval using the blocking method described in Section II-C.

In each case, we performed 50 Monte Carlo simulations corresponding to 50 different initial values of the random source generator. The mean and the standard deviation of SIR over these simulations for $N = 1000$ source samples are shown in Table IV. The bias and standard deviation of coefficients estimated with these two methods are presented in Table V. The results confirm the better performance of the method taking into account the autocorrelation and non-stationarity of signals in nonlinear BSS. The average running time of this method, using a non-optimized Matlab code on a computer with an Intel core 2 Quad CPU with a frequency of 2.8 GHz and a RAM of 4 GB, was about 83 seconds. It should be emphasized that in each iteration of the gradient algorithm, about 68% of the computation time is due to the numerical solution of (45) using a Gauss-Newton algorithm. By optimally implementing this routine (using e.g., a C function), the running time may considerably be reduced. Also, about 31% of the running time is due to the computation of gradient, and only 1% is due to the score function estimation.

Table VI shows the mean of SIR as a function of the sample size N when taking into account the non-stationarity and autocorrelation.

We also performed 50 Monte Carlo simulations corresponding to 50 different values of the nonlinear parameters in the mixing model (45). In each run, these parameters were randomly chosen from a uniform distribution on the interval $[0, 0.8]$. Once more, this choice guarantees the global univalence of the mixing model when the source values are in the interval $[-0.5, 0.5]$. Using the method taking into account the autocorrelation and non-stationarity, the mean and standard deviation of SIR were respectively 30.52 dB and 0.53 dB.

3) Separating Realistic Mixtures of Real-World Spectra: In another test, we first chose three nearly independent real-world spectra $u_1(n), u_2(n), u_3(n)$ from a spectral library compiled by the United States Geological Survey (USGS) [47]. In our tests, 417 wavelengths of each spectrum were used. The reflectance spectra values were in $[0, 1]$. As explained in [43], in hyperspectral remote sensing of non-flat landscapes, the mixing model is linear-quadratic and the mixing coefficients in (32) verify the following constraints: $\sum_k \rho_{ik} = 1$, $\rho_{ik} \geq 0$, $0 \leq \xi_{i(k,l)} \leq 0.5$, $\forall i, k, l$. Thus, in our simulations, we mixed the spectra using the following realistic values which satisfy the above conditions: $\rho_{11} = 0.7$, $\rho_{12} = 0.1$, $\rho_{13} = 0.2$, $\rho_{21} = 0.2$, $\rho_{22} = 0.6$, $\rho_{23} = 0.2$, $\rho_{31} = 0.2$, $\rho_{32} = 0.2$, $\rho_{33} = 0.6$, $\xi_{1(1,2)} = 0.2$, $\xi_{1(1,3)} = 0.3$, $\xi_{1(2,3)} = 0.2$, $\xi_{2(1,2)} = 0.3$, $\xi_{2(1,3)} = 0.1$, $\xi_{2(2,3)} = 0.2$, $\xi_{3(1,2)} = 0.1$, $\xi_{3(1,3)} = 0.3$, $\xi_{3(2,3)} = 0.2$, $\gamma_{ik} = 0$, $\forall i, k$.

Denoting $s_1(n) = \rho_{11}u_1(n)$, $s_2(n) = \rho_{22}u_2(n)$, $s_3(n) = \rho_{33}u_3(n)$, the mixing model can be rewritten as in (33) with $K = 3$ and $b_{ik} = 0$, $\forall i, k$. Using the same method as in Appendix B, it can be checked that the three principal minors of the Jacobian matrix are always positive such that the mixing model is univalent. Finally, we tried to unmix the mixtures using the following four methods:

- Method 1: the well-known FastICA method [48] which is only adapted to linear mixtures and does not take into account the autocorrelation and non-stationarity of spectra.
- Method 2: the well-known blind nonlinear MISEP method [7] which uses neither the knowledge of the parametric model of the mixing transformation nor the structure of signals.
- Method 3: a version of our method which ignores the non-stationarity and the autocorrelation of spectra, i.e., we choose $q = 0$ in (15) and we estimate the marginal score functions on the whole signals.
- Method 4: a version of our method taking into account the structure of spectra, i.e., we choose $q = 1$ in (15) for a first-order Markov model, and we estimate the conditional score functions by splitting the signals in $L = 4$ intervals using the blocking method described in Section II-C. This choice of L results from a trade-off between the number of intervals and the number of samples per interval used for estimating conditional score functions on each interval.

In the last two methods, the linear coefficients were initialized using their estimates provided by linear BSS methods as explained in Section IV-B1. More precisely, we used the non-Markovian ML algorithm described in [30] for initializing Method 3, and the Markovian ML algorithm explained in [4] for initializing Method 4. The nonlinear parameters were initialized to random values in $[0, 1]$. Table VII shows the SIR obtained by these methods and Fig. 5 compares the original and estimated signals. The results are shown after applying some post-processing to obtain zero-mean, unit-variance signals. This post-processing is not necessary for our methods and was essentially performed in order to compare our results with those of FastICA which provides centered and normalized signals at its outputs. As can be seen, the last method provides the best results, which confirms the usefulness of taking into account the nonlinear terms and the structure of model and spectra in the unmixing procedure. For example, the other three methods provide a false peak in the first estimated spectrum (just before the 300th sample) and in the second estimated spectrum (just before or after the 300th sample) while these false peaks do not appear in the spectra estimated by the last method. It is worth mentioning that the constraints on mixing coefficients (especially the sum-to-one constraint on linear coefficients) were not used in our unmixing methods, except a partial use of the non-negativity constraint. In fact, in our methods, at each iteration of the gradient algorithm, we replaced the estimated negative coefficients by zero.

It should also be emphasized that the unmixing using our method is possible only when the source spectra are independent and the mixing model is bijective in their variation domain. In practice, real-world remote sensing spectra are not often statistically independent, especially when they correspond to sim-

TABLE V

BIAS AND STANDARD DEVIATION OF ESTIMATED MIXING COEFFICIENTS IN THE SECOND SIMULATION WITH BILINEAR MODEL. (A) REFERS TO THE BSS METHOD IGNORING THE NON-STATIONARITY AND THE AUTOCORRELATION OF SIGNALS WHILE (B) REFERS TO THE METHOD TAKING THESE PROPERTIES INTO ACCOUNT

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
θ_i	1.00	-1.00	-1.00	-1.00	1.00	1.00	0.80	0.70	0.60	0.50	0.80	0.70	0.80	0.40	0.30
bias($\hat{\theta}_i$)-A	0.03	0.34	0.00	0.24	-0.12	-0.28	-0.63	-0.48	-0.23	-0.44	-0.56	-0.45	-0.04	-0.37	0.16
bias($\hat{\theta}_i$)-B	0.00	0.01	-0.01	-0.01	0.00	0.00	-0.03	-0.03	0.03	0.03	0.00	-0.01	-0.01	-0.03	0.02
std($\hat{\theta}_i$)-A	0.52	0.48	0.57	0.44	0.46	0.46	0.36	0.32	0.50	0.33	0.38	0.53	0.34	0.41	0.40
std($\hat{\theta}_i$)-B	0.04	0.07	0.03	0.05	0.02	0.05	0.11	0.12	0.19	0.11	0.11	0.18	0.09	0.11	0.18

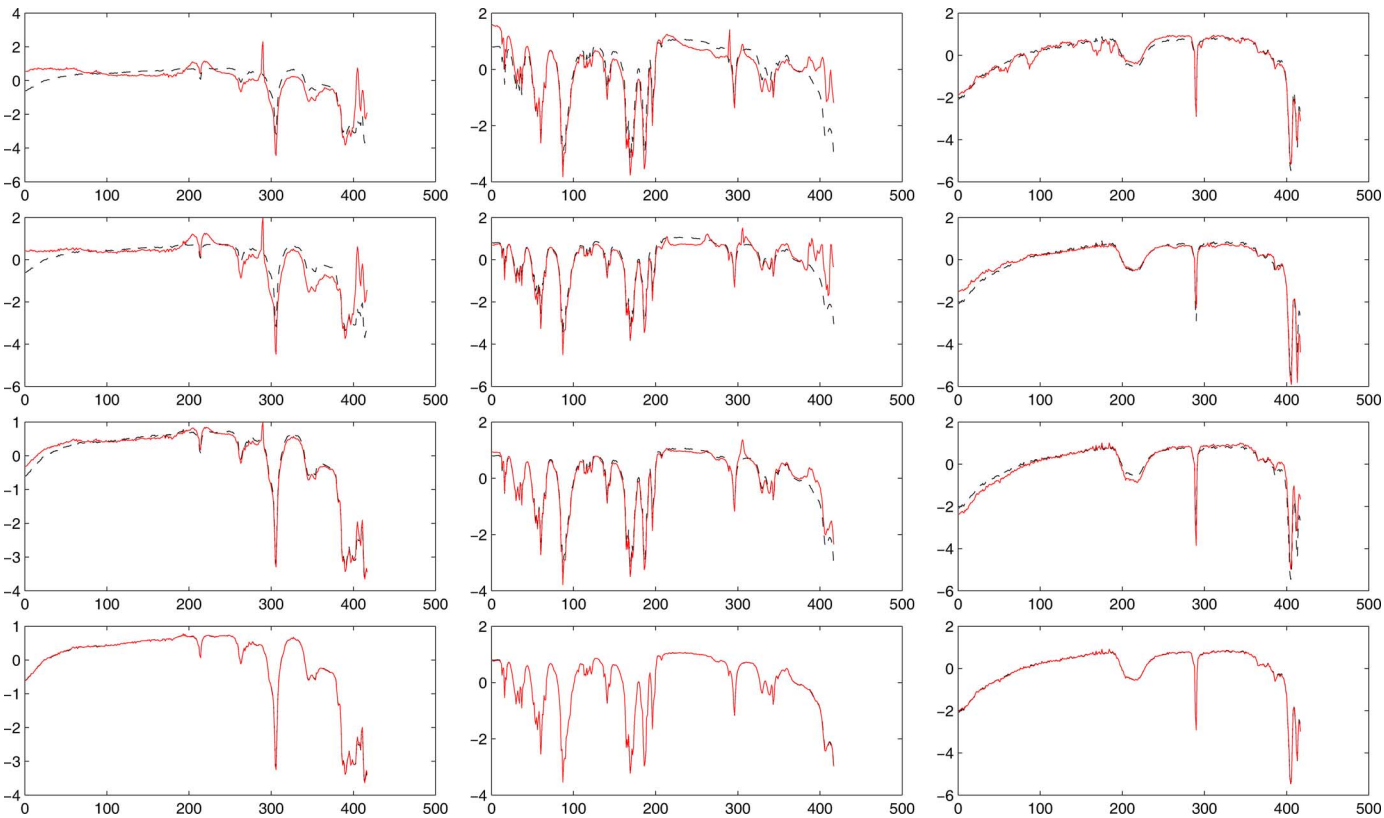

Fig. 5. Dashed lines: original spectra, solid lines: estimated spectra. First row: method 1 (FastICA), second row: method 2 (MISEP), third row: method 3 (this paper), fourth row: method 4 (this paper). Each column corresponds to one spectrum.

TABLE VI

MEAN OF SIR AS A FUNCTION OF SAMPLE SIZE N WITH TAKING INTO ACCOUNT THE NON-STATIONARITY AND AUTOCORRELATION OF SOURCES

N	200	500	1000
mean(SIR) in dB	20.88	26.90	30.69

TABLE VII

SIR OBTAINED IN TESTS USING REAL-WORLD SPECTRA

Method 1	Method 2	Method 3	Method 4
8.50 dB	8.05 dB	14.96 dB	34.61 dB

ilar materials. Moreover, the mixture being usually over-determined, it must first be reduced to a determined mixture as mentioned at the beginning of Section II.

V. CONCLUSION

In this article, we proposed a new method for separating nonlinear mixtures of independent signals. Our method, formulated in a maximum likelihood framework, exploits on the one hand the knowledge of the parametric model of the mixing transformation, and on the other hand the possible autocorrelation and non-stationarity of signals. Its implementation does not require the knowledge of the explicit inverse of the mixing model. Our simulation results using a bijective toy model and a bilinear model showed that taking into account the non-linearity of the mixing transformation and the structure of signals considerably improves the performance of BSS. We also showed in our simulations with the toy model that a good estimation of score functions is critical to achieve BSS.

Several possibilities may be proposed to continue this work. At first, in most of the development of our method, we did not use the possible information about the sources or mixing parameters. For example, in the case of linear-quadratic mixtures of real-world spectra in urban spectral unmixing, it is well known that the sources and mixing parameters are non-negative and their domains of variation are usually known. Moreover, the sum of linear mixing parameters is equal to one. Such

constraints may be applied either implicitly, using e.g., a projected gradient method, or explicitly by adding some regularization terms to the cost function.

The separability issue of nonlinear ICA is another interesting subject. As shown in [49], taking into account the time structure may reduce the indeterminacies involved in nonlinear ICA. However, it is worth studying this issue for each mixing transformation to determine the conditions under which ICA gives a unique solution up to classical indeterminacies. In the case of linear-quadratic models, the separability issue have been addressed in some special configurations in [50] and [21].

Finally, it would be interesting to apply our proposed ML method to other real-world nonlinear mixing models and possibly to real-world data, although the performance evaluation in the latter case requires the knowledge of the actual sources which are generally unknown. Moreover, in this case the observation noise can also have an impact on the performance of the proposed method.

Appendix A
Derivation of Equation (29)

From (24) and using (17), we have

$$\frac{\partial \hat{\mathbf{s}}(n)}{\partial \hat{\theta}} = -\begin{pmatrix} \hat{s}_2(n) & -\hat{s}_1(n) \end{pmatrix} \begin{pmatrix} 3\hat{s}_1(n)^2 & -\hat{\theta} \\ \hat{\theta} & 1 \end{pmatrix}^{-1} = \frac{-1}{\hat{J}(n)} \begin{pmatrix} \hat{s}_2(n)+\hat{\theta}\hat{s}_1(n) & \hat{\theta}\hat{s}_2(n)-3\hat{s}_1(n)^3 \end{pmatrix}. \quad (47)$$

Moreover, from (28) and (47):

$$\frac{\partial \hat{J}(n)}{\partial \hat{\theta}} = 6\hat{s}_1(n)\frac{\partial \hat{s}_1(n)}{\partial \hat{\theta}} + 2\hat{\theta} = 6\hat{s}_1(n)\frac{-(\hat{s}_2(n)+\hat{\theta}\hat{s}_1(n))}{\hat{J}(n)} + 2\hat{\theta}. \quad (48)$$

Inserting the above equations in (15), by choosing $q = 0$ and considering the stationary case where score functions do not depend on n, leads to (29).

Appendix B
Global Univalence of the Mixture Used in the Simulations of Section IV-B1

According to the fundamental global univalence theorem [51], a differential mapping $F : \Omega \subset \mathbb{R}^K \rightarrow \mathbb{R}^K$, where Ω is a rectangular region in $\mathbb{R}^K$, is globally univalent in Ω if its Jacobian matrix at $\mathbf{s}$ is a P-matrix[7] for every $\mathbf{s} \in \Omega$. Considering the mixing parameter values θ_i used in our simulations of Section IV-B1 with the bilinear mixture described by (45), and using (34), the Jacobian matrix reads (omitting the index n due to space limitation):

$$\mathcal{J} = \begin{pmatrix} 1+0.8s_2+0.7s_3 & -1+0.5s_2+0.8s_3 & 1+0.8s_2+0.4s_3 \\ 1+0.8s_1+0.6s_3 & 1+0.5s_1+0.7s_3 & 1+0.8s_1+0.3s_3 \\ -1+0.7s_1+0.6s_2 & -1+0.8s_1+0.7s_2 & 1+0.4s_1+0.3s_2 \end{pmatrix}, \quad (49)$$

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

and its three principal minors are

$$\begin{aligned} minor_{11} &= (1+0.5s_1+0.7s_3)(1+0.4s_1+0.3s_2) \\ &\quad -(1+0.8s_1+0.3s_3)(-1+0.8s_1+0.7s_2) \\ minor_{22} &= (1+0.8s_2+0.7s_3)(1+0.4s_1+0.3s_2) \\ &\quad -(1+0.8s_2+0.4s_3)(-1+0.7s_1+0.6s_2) \\ minor_{33} &= (1+0.8s_2+0.7s_3)(1+0.5s_1+0.7s_3) \\ &\quad -(-1+0.5s_2+0.8s_3)(1+0.8s_1+0.6s_3). \end{aligned} \quad (50)$$

Since s_1, s_2 and s_3 belong to $[-0.5, 0.5]$, it can easily be verified that the first term in each of the above expressions is always positive and the second term is always negative so that the three minors are always positive over $[-0.5, 0.5]^3$.

References

[1] *Handbook of Blind Source Separation. Independent Component Analysis and Applications*, P. Comon and C. Jutten, Eds.. Oxford, U.K.: Academic, 2010.

[2] J.-F. Cardoso, "The three easy routes to independent component analysis, contrasts and geometry," in *Proc. ICA*, San Diego, CA, USA, 2001, pp. 1–6.

[3] S. Hosseini, C. Jutten, and D. T. Pham, "Markovian source separation," *IEEE Trans. Signal Process.*, vol. 51, no. 12, pp. 3009–3019, Dec. 2003.

[4] R. Guidara, S. Hosseini, and Y. Deville, "Blind separation of non-stationary Markovian sources using an equivariant Newton-Raphson algorithm," *IEEE Signal Process. Lett.*, vol. 16, no. 5, pp. 426–429, May 2009.

[5] H. Lappalainen and A. Honkela, "Bayesian non-linear independent component analysis by multi-layer perceptrons," in *Adv. Ind. Compon. Anal.*, M. Girolami, Ed. New York, NY, USA: Springer-Verlag, 2000, pp. 93–121.

[6] Y. Tan, J. Wang, and J. Zurada, "Nonlinear blind source separation using a radial basis function network," *IEEE Trans. Neural Netw.*, vol. 12, pp. 124–134, 2001.

[7] L. B. Almeida, "MISEP – linear and nonlinear ICA based on mutual information," *J. Mach. Learn. Res.*, vol. 4, pp. 1297–1318, 2003.

[8] S. Harmeling, A. Ziehe, B. Blankertz, and K.-R. Müller, "Kernel-based nonlinear blind source separation," *Neural Comput.*, vol. 15, pp. 1089–1124, 2003.

[9] A. Honkela, H. Valpola, A. Ilin, and J. Karhunen, "Blind separation of nonlinear mixtures by variational Bayesian learning," *Digit. Signal Process.*, vol. 17, pp. 914–934, 2007.

[10] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Netw.*, vol. 12, no. 3, pp. 429–439, 1999.

[11] A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2807–2820, 1999.

[12] A. Taleb, "A generic framework for blind source separation in structured nonlinear models," *IEEE Trans. Signal Process.*, vol. 50, no. 8, pp. 1819–1830, Aug. 2002.

[13] C. Jutten, B. Babaie-Zadeh, and S. Hosseini, "Three easy ways for separating nonlinear mixtures?," *Signal Process.*, vol. 84, no. 2, pp. 217–229, Feb. 2004.

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

[15] S. Achard, D. Pham, and C. Jutten, "Blind source separation in post nonlinear mixtures," in *Proc. ICA*, San Diego, CA, USA, 2001, pp. 259–300.

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

[17] 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*, Atlanta, GA, USA, May 1996, pp. 2718–2721.

[18] S. Hosseini and Y. Deville, "Blind separation of linear-quadratic mixtures of real sources using a recurrent structure," in *Proc. 7th Int. Workshop-Conf. Artif. Neural Netw. (IWANN)*, Mao, Menorca, Spain, Jun. 2003, vol. 2, pp. 241–248.
[19] S. Hosseini and Y. Deville, "Blind maximum likelihood separation of a linear-quadratic mixture," in *Proc. ICA*, Granada, Sep. 2004, pp. 694–701.
[20] M. Castella, "Inversion of polynomial systems and separation of nonlinear mixtures of finite-alphabet sources," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pt. 2, pp. 3905–3917, Aug. 2008.
[21] M. S. C. Almeida and L. B. Almeida, "Nonlinear separation of show-through image mixtures using a physical model trained with ICA," *Signal Process.*, vol. 92, no. 4, pp. 872–884, Apr. 2012.
[22] J. Eriksson and V. Koivunen, "Blind identifiability of class of nonlinear instantaneous ICA models," in *Proc. EUSIPCO*, Toulouse, France, Sep. 2002, vol. 2, pp. 7–10.
[23] G. Bedoya, "Non-linear blind signal separation for chemical solid-state sensor arrays," PhD, Dept. of Electrical Eng., Technical Univ. of Catalonia, Barcelona, Spain, 2006.
[24] L. T. Duarte and C. Jutten, "Blind source separation of a class of nonlinear mixtures," in *Proc. ICA*, London, U.K., Sep. 2007, pp. 41–48.
[25] Y. Deville, "ICA-based and second-order separability of nonlinear models involving reference signals : General properties and application to quantum bits," *Signal Process.*, vol. 92, no. 8, pp. 1785–1795, Aug. 2012.
[26] T. Blaschke, T. Zito, and L. Wiskott, "Independent slow feature analysis and nonlinear blind source separation," *Neural Comput.*, vol. 19, pp. 994–1021, 2007.
[27] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Muller, "Separation of post-nonlinear mixtures using ACE and temporal decorrelation," in *Proc. ICA*, San Diego, CA, USA, 2001, pp. 433–438.
[28] A. Ziehe, M. Kawanabe, and S. Harmeling, "Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation," *J. Mach. Learn. Res.*, vol. 4, pp. 1319–1338, 2003.
[29] M. Gaeta and J.-L. Lacoume, "Source separation without prior knowledge: The maximum likelihood solution," in *Proc. EUSIPCO*, 1990, pp. 621–624.
[30] D.-T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1712–1725, 1997.
[31] J.-F. Cardoso, "Infomax and maximum likelihood for source separation," *IEEE Signal Process. Lett.*, vol. 4, pp. 112–114, 1997.
[32] P. Comon, "Independent component analysis—A new concept?," *Signal Process.*, vol. 36, pp. 287–314, 1994.
[33] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proc. IEEE*, vol. 86, pp. 2009–2025, 1998.
[34] M. Babaie-Zadeh and C. Jutten, "A general approach for mutual information minimization and its application to blind source separation," *Signal Process.*, vol. 85, pp. 975–995, 2005.
[35] S. Hosseini and Y. Deville, "Recurrent networks for separating extractable-target nonlinear mixtures. Part II. Blind configurations," *Signal Process.*, vol. 93, no. 4, pp. 671–683, Apr. 2013.
[36] M. Babaie-Zadeh, "On blind source separation in convolutive and nonlinear mixtures," Ph.D. dissertation, LIS-INPG, Grenoble, France, 2002.
[37] D.-T. Pham, "Blind separation of non stationary non Gaussian sources," in *Proc. 11th Eur. Signal Process. Conf. (EUSIPCO)*, Toulouse, France, Sep. 2002, pp. 67–70.
[38] Y. Deville and S. Hosseini, "Recurrent networks for separating extractable-target nonlinear mixtures. Part I : Non-blind configurations," *Signal Process.*, vol. 89, no. 4, pp. 378–393, Apr. 2009.
[39] 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," presented at the EUSIPCO, Lausanne, Switzerland, Aug. 2008.
[40] 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 Sens. Environ.*, vol. 113, pp. 1183–1193, 2009.
[41] 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," *Internat. J. Remote Sens.*, vol. 30, no. 11, pp. 2951–2962, Jun. 2009.
[42] J. Nascimento and J. Bioucas-Dias, "Nonlinear mixture model for hyperspectral unmixing," in *Proc. SPIE Conf. Image Signal Process. Remote Sens.*, 2009, vol. SPIE-7477.
[43] I. Meganem, P. Deliot, X. Briottet, Y. Deville, and S. Hosseini, "Linear–quadratic mixing model for reflectances in urban environments," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 544–558, Jan. 2014.
[44] P. Gader, D. Dranishnikov, A. Zare, and J. Chanussot, "A sparsity promoting bilinear unmixing model," presented at the 4th Workshop Hyperspectral Image Signal Process. (WHISPERS), Shanghai, China, Jun. 2012.
[45] N. Yokoya, J. Chanussot, and A. Iwasaki, "Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1430–1437, Feb. 2014.
[46] K. B. Petersen and M. S. Pedersen, The Matrix Cookbook Tech. Univ. of Denmark, 2012 [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?327
[47] R. N. Clark, G. A. Swayze, R. Wise, E. Livo, T. Hoefen, R. Kokaly, and S. J. Sutley, 2007, USGS Digital Spectral Library Splib06a: U.S. Geological Survey 2007, vol. 231, Digital Data Series [Online]. Available: http://speclab.cr.usgs.gov/spectral-lib.html
[48] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, 1997.
[49] S. Hosseini and C. Jutten, "On the separability of nonlinear mixtures of temporally correlated sources," *IEEE Signal Process. Lett.*, vol. 10, no. 2, pp. 43–46, Feb. 2003.
[50] F. J. Theis and W. Nakamura, "Quadratic independent component analysis," *IEICE Trans. Fundam.*, vol. E87-A, no. 9, pp. 2355–2363, 2004.
[51] T. Parthasarathy, "On global univalence theorems," in *Lecture Notes in Mathematics*. New York, NY, USA: Springer-Verlag, 1983, vol. 977.

Shahram Hosseini was born in Shiraz, Iran, in 1968. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 1991 and 1993, respectively, and the Ph.D. degree in signal processing from the Institut National Polytechnique, Grenoble, France, in 2000. He is currently an Associate Professor at the Université Paul Sabatier Toulouse 3, Toulouse, France. His research interests include blind source separation, artificial neural networks, and adaptive signal processing.

Yannick Deville (M'98) was born in Lyon, France, in 1964. He graduated from the Ecole Nationale Supérieure des Télécommunications de Bretagne (Brest, France) in 1986. He received the D.E.A. and Ph.D. degrees, both in Microelectronics, from the University of Grenoble (France), in 1986 and 1989 respectively. From 1986 to 1997, he was a Research Scientist at Philips Research Labs (Limeil, France). His investigations during this period concerned various fields, including GaAs integrated microwave RC active filters, VLSI cache memory architectures and replacement algorithms, neural network algorithms and applications, and nonlinear systems. Since 1997, he has been a Professor at the University of Toulouse (France). From 1997 to 2004, he was with the Acoustics lab of that University. Since 2004, he has been with the Astrophysics lab in Toulouse, which is part of the University and of the French National Center for Scientific Research (CNRS). His current major research interests include signal and image processing, higher-order statistics, time-frequency analysis, neural networks, and especially Blind Source Separation methods (including Independent or Sparse Component Analysis) and their applications to remote sensing, astrophysics, quantum information processing, acoustics and communication/electromagnetic signals.