Correspondence

Fast Blind Separation of Long Mixture Recordings Using Multivariate Polynomial Identification

Johan Thomas, Yannick Deville, and Shahram Hosseini

***Abstract*—This correspondence presents new approaches for optimizing kurtosis-based separation criteria in the case of long mixture recordings. Our methods are based on a multivariate polynomial identification step that avoids the computation of signal statistics at each step of the commonly used fixed-point optimization algorithms. As compared to the well-known FastICA algorithm and to our recent DFICA algorithm intended for blind partial separation of nonstationary sources, our new methods are very computationally efficient for long recordings of a moderate number of mixed sources. They are therefore especially suited to blind image separation, because of the high number of pixels in light sensors. Our algorithms also avoid the computation and storage of the sphered observation vector, thus saving memory space.**

***Index Terms*—Blind source separation (BSS), independent component analysis (ICA), kurtosis, long mixture recordings, non-Gaussian signals.**

I. Introduction

Blind source separation (BSS) aims at estimating a set of N unobserved source signals from the observation of P mixtures of these signals when the mixture parameters are unknown. Let us denote by $\mathbf{s}(n) = [s_1(n), \ldots, s_N(n)]^t$ the vector of real-valued zero-mean sources and $\mathbf{x}(n) = [x_1(n), \ldots, x_P(n)]^t$ the vector of observations. We here consider the case of linear instantaneous mixtures which occur when each observation is a linear combination of the original sources. By denoting $\mathbf{A}$ the $P \times N$ scalar mixing matrix containing the real-valued scale coefficients between each source and each observation, the relationship between the source and observation vectors reads in matrix form as $\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n)$. Among all BSS methods, we here consider those that use independent component analysis (ICA) [1] and especially which aim at maximizing the non-Gaussianity of some output signals to estimate the sources.

Non-Gaussianity was considered for the first time in ICA by Comon [2], [3] who used a fourth-order statistical parameter, i.e., the non-normalized kurtosis, which is defined for a zero-mean variable $y(n)$ by

$$\mathrm{kurt}_y(n) = E\left\{y^4(n)\right\} - 3E\left\{y^2(n)\right\}^2. \tag{1}$$

Comon proposed to achieve the separation in two stages. The first one performs a sphering process which consists in decorrelating and normalizing the set of observations. The second one optimizes a criterion (COM2) defined as the sum of the squared kurtoses of the output signals which are linear combinations of the sphered observations. Moreau and Macchi then proposed to use the sum of the kurtoses of the output signals as a criterion, which guarantees separation when the kurtoses of the sources have the same sign [4]. Comon and Moreau [5] then derived an analytical solution for the optimization of this criterion (COM1).

Delfosse and Loubaton then proposed a deflation approach which estimates the sources iteratively [6]. They proposed to first perform the same sphering process as above. In the second stage of this approach, the algorithm maximizes the absolute value or square of the non-normalized kurtosis of one linear combination $y(n) = \sum_{i=1}^{N} w(i) z_i(n)$ of the N sphered signals $z_i(n)$ under the constraint $\|\mathbf{w}\| = 1$. This is done by iteratively updating the vector $\mathbf{w}$ with the following two operations: a gradient ascent algorithm step for the criterion $\mathrm{kurt}_y(n)$ and a normalization step to obtain $\|\mathbf{w}\| = 1$. This yields an estimate of one source. The other sources are then estimated by a deflation approach [6].

In the FastICA algorithm [7], the same sphering process is applied, but FastICA then optimizes the above kurtotic criterion by using a fixed-point algorithm instead of a gradient ascent approach. The fixed-point iteration alternately updates the extraction vector with the gradient of the kurtosis and then normalizes this vector. This algorithm takes advantage of the fact that when the extraction vector $\mathbf{w}$ maximizes the absolute value of the kurtosis under the constraint $\|\mathbf{w}\| = 1$, the gradient of this kurtosis is collinear to $\mathbf{w}$. As it is fast and has no tunable parameter, the FastICA algorithm is very popular in the ICA community and is used for many real applications in various domains such as astrophysical image separation [8]. Recently, we proposed a differential fast fixed-point algorithm, called DFICA, for underdetermined blind partial separation of some nonstationary sources in presence of stationary sources [9].

Recently, Zarzoso *et al.* [10] proposed an alternative to the FastICA algorithm, called RobustICA, which consists of performing exact line search optimization of a contrast function which is defined as the normalized kurtosis:

$$\mathrm{kurt}_y^{\mathrm{norm}}(n) = \frac{\mathrm{kurt}_y(n)}{E\left\{y^2(n)\right\}^2}. \tag{2}$$

In this paper, we propose an approach for optimizing the same criterion as in FastICA in order to achieve the blind separation of (over)determined mixtures. This approach can be applied to the underdetermined case by using the concept of differential source separation. In the (over)determined case ($P \geq N$), we make the following classical assumptions concerning the mixture model:

- $\mathbf{s}(n)$ is stationary and spatially independent, i.e., its components $s_i(n)$ are statistically independent from each other; we also assume that at most one of these components is Gaussian;
- the scalar mixing matrix $\mathbf{A}$ is full column rank. If $P > N$, we can use Principal Component Analysis (PCA) to reduce the number of observations to N.

We also assume that the observations contain a high number T of samples (typically more than 100 000 samples) so that the FastICA and DFICA algorithms yield high processing times for estimating statistics at each step of the kurtosis-based optimization algorithms.[1] This especially concerns image sources: because of the present size increase of CCD and CMOS-type light sensors which imply large numbers of samples, our proposed approach should then be more efficient than the classical FastICA and DFICA algorithms. Our method is also attractive for audio source separation when the signals are stationary over

Manuscript received November 19, 2007; revised July 5, 2008. First published August 19, 2008; current version published October 15, 2008. The associate editor coordinating the review of this paper and approving it for publication was Dr. Deniz Erdogmus.

The authors are with the Laboratoire d'Astrophysique de Toulouse-Tarbes—Université de Toulouse/CNRS, 31400 Toulouse, France (e-mail: jthomas@ast.obs-mip.fr; ydeville@ast.obs-mip.fr; shosseini@ast.obs-mip.fr).

Digital Object Identifier 10.1109/TSP.2008.929666

[1]The RobustICA algorithm does not explicitly tackle the problems encountered with long signal recordings as the complexity of each iteration stays proportional to the number of samples, even though less iterations are required compared to the FastICA algorithm.

long enough periods, in data compression when the files to be compressed are large, or in telecommunications. Besides, we suppose that the sources to be estimated are not too numerous (typically up to seven sources), which is often the case.

This paper is organized as follows. In Section II, after describing the FastICA method and its limitations, we propose our own approach that aims at improving the separation algorithm in the case of long mixture recordings. We also stress the possibility of extending our algorithm to the underdetermined case by applying the differential source separation principle. The experimental section then compares the speeds of our optimized algorithms with the FastICA approach and our recent differential DFICA method [9]. Conclusions are drawn from this investigation in Section IV.

II. An Optimized Fast Fixed-Point Algorithm For (Over)Determined BSS

A. Principles and Limitations of Standard FastICA Approach

The first step of the FastICA algorithm consists in decorrelating and normalizing the observations, which can be done by the following operation:

$$\mathbf{z}(n) = \mathbf{B}\mathbf{x}(n) \tag{3}$$

where $\mathbf{B}$ is the sphering matrix of the observation vector and $\mathbf{z}(n)$ is the sphered observation vector, whose correlation matrix is equal to the identity matrix ($\mathbf{R_z} = \mathbf{I}_N$). The sphering matrix is generally obtained by means of the eigendecomposition of the correlation matrix $\mathbf{R_x}$, i.e., $\mathbf{B}$ is taken equal to $\mathbf{\Delta}^{-1/2}\mathbf{E}^t$, where $\mathbf{E}$ and $\mathbf{\Delta}$ are respectively the orthogonal matrix and the diagonal matrix associated with the eigendecomposition of $\mathbf{R_x}$. This process implies equality between the power of the subsequent output signal $y(n) = \mathbf{w}^t\mathbf{z}(n)$ and the squared norm of the extraction vector $\mathbf{w}$.

There exist two versions of the FastICA second stage. The parallel version, whose convergence has been rigorously proved by Oja and Yuan in [11], extracts all the sources at the same time by repeatedly updating a set of extraction vectors with the gradients of the kurtoses, orthogonalizing and normalizing these vectors by means of the symmetrical operation $\mathbf{W} = \mathbf{W}\left(\mathbf{W}^t\mathbf{W}\right)^{-1/2}$, where the matrix $\mathbf{W}$ is composed of the N extraction vectors $\mathbf{w}_i$ arranged column-wise. The fixed-point iteration updates each of the N extraction vectors by the gradients of the kurtosis, which reads

$$\forall\, i = 1 \ldots N$$
$$\mathbf{w}_i \leftarrow E\left\{\mathbf{z}(n)\left(\mathbf{w}_i^t\mathbf{z}(n)\right)^3\right\} - 3\mathbf{w}_i \propto \frac{\partial \mathrm{kurt}_{\mathbf{w}_i^t\mathbf{z}}(n)}{\partial \mathbf{w}_i}. \tag{4}$$

This is performed in the FastICA package [12] by the operation

$$\mathtt{W} = (\mathtt{Z} * ((\mathtt{Z}' * \mathtt{W}).\hat{}\,3))/\mathtt{T} - 3 * \mathtt{W} \tag{5}$$

where $\mathtt{Z}$ is an $N \times T$ matrix containing the T samples of the sphered observations and $\mathtt{W}$ implements the above $N \times N$-size matrix $\mathbf{W}$. As update (5) requires to store $\mathtt{Z}$ and $(\mathtt{Z}' * \mathtt{W}).\hat{}\,3$, we thus need $2N \times T$ memory words. It may be shown[2] easily that operation (5) requires to compute $(4N^2 + N)T + 2N^2$ operations. It was pointed out in [13] that the averages in the update equation could possibly be estimated by using a smaller sample size, thus reducing the computational load. However, this reduced number of samples is also said to have a considerable effect on the accuracy of the final estimate of $\mathbf{W}$. Since we here aim at achieving good accuracy, we use all available data.

The deflation version of FastICA extracts the sources one by one, by applying a deflation-based orthogonalization procedure instead of the above mentioned symmetrical one. In this case, at each step of the optimization, a unique vector $\mathbf{w}$ is updated with the gradient of the kurtosis of $\mathbf{w}^t\mathbf{z}(n)$ (the matrix $\mathtt{W}$ is replaced by a unique N-size column vector $\mathtt{w}$) and the resulting vector is projected on the space orthogonal to the previously estimated extraction vectors. This is done by the operation $\mathbf{w} = \mathbf{w} - \mathbf{W}\mathbf{W}^t\mathbf{w}$, where $\mathbf{W}$ contains the previously estimated vectors arranged column-wise. The resulting vector is then normalized by the operation $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$.

B. A New Approach for Optimizing Kurtosis

In this section, we propose a method to avoid the above important memory requirement and the computation of the sample statistics in each update operation of the optimization of $\mathbf{W}$ or $\mathbf{w}$. This method is based on a multivariate polynomial identification that allows us to perform the computations of the above fixed-point algorithms in a space of polynomial coefficients. It also avoids the computation and the storage of the sphered observation vector $\mathbf{z}(n)$ defined by (3). This new approach is attractive for most configurations (typically with $N \leq 7$) in terms of computation time and requested memory space, particularly for long mixture recordings, whereas it converges towards exactly the same points as the standard FastICA algorithms.

Let us denote by $y(n) = \mathbf{w}^t\mathbf{z}(n)$ a linear combination of the entries of the sphered observation vector $\mathbf{z}(n)$ used in the FastICA algorithm and computed by using (3) (the computation of $\mathbf{z}(n)$ will be avoided in our approach as explained further). Since all signals are assumed to be zero-mean, the kurtosis of $y(n)$ reads

$$\mathrm{kurt}_y(n) = \mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n) = E\left\{(\mathbf{w}^t\mathbf{z}(n))^4\right\} - 3E\left\{(\mathbf{w}^t\mathbf{z}(n))^2\right\}^2. \tag{6}$$

As we have

$$\begin{aligned} E\left\{(\mathbf{w}^t\mathbf{z}(n))^2\right\} &= E\left\{\mathbf{w}^t\mathbf{z}(n)\mathbf{z}^t(n)\mathbf{w}\right\} \\ &= \mathbf{w}^t E\left\{\mathbf{z}(n)\mathbf{z}^t(n)\right\}\mathbf{w} \\ &= \mathbf{w}^t\mathbf{I}\mathbf{w} \\ &= \|\mathbf{w}\|^2 \end{aligned} \tag{7}$$

(6) becomes

$$\begin{aligned} \mathrm{kurt}_y(n) &= E\left\{\sum_{i_1,\ldots,i_4=1}^{N}\prod_{k=1}^{4} w(i_k)z_{i_k}(n)\right\} - 3\left(\sum_{i=1}^{N} w(i)^2\right)^2 \\ &= \sum_{i_1,i_2,i_3,i_4=1}^{N}\left(\prod_{k=1}^{4} w(i_k)\right)E\left\{\prod_{k=1}^{4} z_{i_k}(n)\right\} \\ &\quad - 3\sum_{i_1,i_2=1}^{N} w(i_1)^2 w(i_2)^2. \end{aligned} \tag{8}$$

As (8) is a fourth-order polynomial with respect to the variables $w(1), \ldots, w(N)$, we can define a set of coefficients $(\alpha_{\mathbf{d}})_{\mathbf{d}\in D}$ with $D = \left\{\mathbf{d} \in \{0,\ldots,4\}^N \setminus \sum_{k=1}^{N} d(k) = 4\right\}$ such that

$$\mathrm{kurt}_y(n) = \sum_{\mathbf{d}\in D}\alpha_{\mathbf{d}}\prod_{k=1}^{N} w(k)^{d(k)}. \tag{9}$$

Equation (9) corresponds to the canonical expansion[3] of the kurtosis with respect to the variables $w(1), \ldots, w(N)$. Let us denote by R

[2]We assume that each matrix product $\mathbf{M}\mathbf{N}$, if the dimensions of $\mathbf{M}$ and $\mathbf{N}$ are respectively $d_1 \times d_2$ and $d_2 \times d_3$, needs $d_1 d_3(2d_2 - 1)$ elementary operations.

[3]The coefficients $\alpha_{\mathbf{d}}$ here include the binomial coefficients which result from the expansion of the fourth power of $y(n)$.

the cardinality[4] of D, by $\mathbf{d}_1, \ldots, \mathbf{d}_R$ its elements and $\alpha_1, \ldots, \alpha_R$ the values of $(\alpha_{\mathbf{d}})_{\mathbf{d}\in D}$. We have then

$$\mathrm{kurt}_y(n) = \sum_{r=1}^{R} \alpha_r \prod_{k=1}^{N} w(k)^{d_r(k)}. \tag{10}$$

It would be possible to directly compute the coefficients α_r which depend on fourth-order cross moments of the signals $z_1(n), \ldots, z_N(n)$. We here propose a method which is more efficient in terms of computation time. Let us denote by $(\mathbf{v}_i)_{i=1..R}$ a family of R different N-size vectors. Replacing $\mathbf{w}$ by $\mathbf{v}_i$ in (10), we then have

$$\forall\, i = 1 \ldots R, \quad \mathrm{kurt}_{\mathbf{v}_i^t\mathbf{z}}(n) = \sum_{j=1}^{R} \alpha_j \prod_{k=1}^{N} v_i(k)^{d_j(k)} \tag{11}$$

which can be expressed by the matrix relation

$$\mathbf{M}\boldsymbol{\alpha} = \mathbf{k} \tag{12}$$

where $\boldsymbol{\alpha}$ is the column vector of coefficients $(\alpha_r)_{r=1\ldots R}$ and where the matrix $\mathbf{M} = [m_{ij}]_{i,j=1\ldots R}$ and the column vector $\mathbf{k} = (k_i)_{i=1\ldots R}$ are respectively defined by

$$\begin{cases} \forall\, i,j = 1 \ldots R, & m_{ij} = \prod_{k=1}^{N} v_i(k)^{d_j(k)} \\ \forall\, i = 1 \ldots R, & k_i = \mathrm{kurt}_{\mathbf{v}_i^t\mathbf{z}}(n). \end{cases} \tag{13}$$

By choosing R extraction vectors $\mathbf{v}_i$ of size N which yield a nonsingular matrix $\mathbf{M}$, we can thus identify the R coefficients $(\alpha_r)_{r=1\ldots R}$ by computing the kurtoses k_i of the R signals $\mathbf{v}_i^t\mathbf{z}(n)$ obtained with these extraction vectors and by using the inverse relation of (12), i.e., $\boldsymbol{\alpha} = \mathbf{M}^{-1}\mathbf{k}$. In the Appendix, we propose particular vectors $(\mathbf{v}_i)$ which yield a well-conditioned matrix $\mathbf{M}$. It must be noted that thanks to the sphering process, $\forall\, i = 1 \ldots R$, $E\left\{\left(\mathbf{v}_i^t\mathbf{z}(n)\right)^2\right\} = \|\mathbf{v}_i\|^2$ as proved by (7) and thus $\mathrm{kurt}_{\mathbf{v}_i^t\mathbf{z}}(n) = E\left\{\left(\mathbf{v}_i^t\mathbf{z}(n)\right)^4\right\} - 3\|\mathbf{v}_i\|^4$ so that we only have to compute a single fourth-order moment for each of the R extraction vectors. Furthermore, the relation

$$\begin{aligned} \forall\, i = 1 \ldots R, \quad \mathbf{v}_i^t\mathbf{z}(n) &= \mathbf{v}_i^t\boldsymbol{\Delta}^{-1/2}\mathbf{E}^t\mathbf{x}(n) \\ &= \mathbf{u}_i^t\mathbf{x}(n) \end{aligned} \tag{14}$$

with $\mathbf{u}_i^t = \mathbf{v}_i^t\boldsymbol{\Delta}^{-1/2}\mathbf{E}^t$ allows us to avoid the computation of the sphered observation vector $\mathbf{z}(n)$. In the following, $\mathbf{U}$ and $\mathbf{V}$ denote the matrices containing respectively the sets of vectors $(\mathbf{u}_i)_{i=1\ldots R}$ and $(\mathbf{v}_i)_{i=1\ldots R}$ arranged column-wise.

Let us note that the matrix $\mathbf{M}$ is independent from the sources and from the mixing matrix. It is then possible to compute its inverse once for all for the set of vectors $(\mathbf{v}_i)_{i=1\ldots R}$ that has been chosen for the considered number of sources. After we have identified the set of coefficients $(\alpha_r)_{r=1\ldots R}$ by computing the kurtoses k_i of the R signals associated with the R vectors $(\mathbf{v}_i)_{i=1\ldots R}$ and by deriving $\boldsymbol{\alpha} = \mathbf{M}^{-1}\mathbf{k}$, we aim at maximizing the absolute value of $\mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n) = \sum_{r=1}^{R} \alpha_r \prod_{k=1}^{N} w(k)^{d_r(k)}$ with respect to $\mathbf{w}$ under the constraint $\|\mathbf{w}\| = 1$. This can be done in the same way as in the standard FastICA algorithm but in the space of polynomial coefficients α_r involved in (10), instead of in the space of signals associated to (6). As in the deflation-based FastICA algorithm, we alternately update the extraction vector with the gradient of the kurtosis and normalize this vector. For a parallel version, N vectors are updated and then orthonormalized. We thus use the gradient of $\mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n)$ with respect to $\mathbf{w}$, which reads

$$\frac{\partial \mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n)}{\partial \mathbf{w}} = \sum_{r=1}^{R} \alpha_r \frac{\partial \prod_{k=1}^{N} w(k)^{d_r(k)}}{\partial \mathbf{w}} \tag{15}$$

[4]It may be shown that $\mathrm{card}(D) = R = N + 3N(N-1)/2 + N(N-1)(N-2)/2 + N(N-1)(N-2)(N-3)/24$.

with

$$\frac{\partial \prod_{k=1}^{N} w(k)^{d_r(k)}}{\partial w(j)} = \begin{cases} d_r(j)w(j)^{d_r(j)-1}\prod_{k\neq j} w(k)^{d_r(k)}, & \text{if } d_r(j) \geq 1 \\ 0, & \text{if } d_r(j) = 0. \end{cases} \tag{16}$$

Let us now summarize the parallel version of our optimized FastICA algorithm, which we call O-FICA.

- Estimate the correlation matrix $\mathbf{R}_{\mathbf{x}}$ of the observation vector $\mathbf{x}(n)$ and compute its eigendecomposition, which yields an orthogonal matrix $\mathbf{E}$ and a diagonal matrix $\boldsymbol{\Delta}$, and compute the sphering matrix $\mathbf{B} = \boldsymbol{\Delta}^{-1/2}\mathbf{E}^t$.
- Compute $\mathbf{U} = \mathbf{B}^t\mathbf{V}$, where $\mathbf{V}$ is composed of the R extraction vectors proposed in the Appendix.
- Compute the kurtoses $(k_i)_{i=1\ldots R}$ of the R signals $\mathbf{u}_i^t\mathbf{x}(n), i = 1 \ldots R$ and determine the set of coefficients $\boldsymbol{\alpha} = (\alpha_r)_{r=1\ldots R}$ by means of the pre-computed inverse of $\mathbf{M}$ defined by (13), with the relation $\boldsymbol{\alpha} = \mathbf{M}^{-1}\mathbf{k}$.
- Initialize $\mathbf{W}$ to N different N-size vectors arranged column-wise, i.e., $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N] = [w_{ji}]$ and repeat until convergence:
  1) $\forall\, i,j = 1 \ldots N, w_{ji} \leftarrow \sum_{r=1}^{R} \alpha_r d_r(j) w_{ji}^{\max(d_r(j)-1,0)} \prod_{k\neq j} w_{ki}^{d_r(k)}$;
  2) $\mathbf{W} \leftarrow \mathbf{W}(\mathbf{W}^t\mathbf{W})^{-1/2}$.

The deflation version of our O-FICA algorithm can be easily derived from the relation between the parallel and deflation versions of the classical FastICA algorithm. We point out that this new algorithm does not need to estimate signal statistics at each step of the fixed-point algorithm, as opposed to the standard FastICA algorithm, but only works with polynomial coefficients α_r, which is more efficient provided that they are not too numerous (this is typically the case for $N \leq 7$). Moreover, we avoid to store the matrices Z and (Z' * W).^3 during the fixed-point algorithm, which represents $2NT$ memory words. It may be shown that we only have to store $R^2 + R + RN + N^2$ memory words during the optimization, which is independent from the number of samples T and therefore especially attractive for a high T.

Tables I and II present the number of floating-point operations needed by the FastICA and O-FICA algorithms, respectively.[5] These tables show that the number of elementary operations of one FastICA iteration is approximately proportional to the number of samples T, whereas O-FICA iteration complexity is proportional to R, which is independent from and here much lower than T. Besides, the FastICA initialization stage, which comes before the optimization iterations, is faster than the O-FICA one. However, this criterion is not the only one to be considered to assess the speed of an algorithm. Indeed, in modern computer architectures, the speed of execution strongly depends on the possibility or not to store intermediate results in cache memory. The interest of our O-FICA algorithm is increased by the fact that, unlike FastICA, it does not require to manipulate large-size matrices that cannot be stored in cache memory. In the experimental section, we will determine the computation gain of our optimized approach by doing some statistical tests with the Matlab software.

C. Extension to Underdetermined Blind Source Separation

In this subsection, we briefly present the application of the above polynomial identification principle to our differential fast fixed-point algorithm DFICA recently proposed in [9] (we call O-DFICA the optimized version of our DFICA algorithm).

[5]Because of the assumption $N < 7$, the eigendecomposition of $\mathbf{R}_{\mathbf{x}}$ and the computation of $\mathbf{B}$ are very fast; their number of operations is indeed proportional to N^3, which is denoted $O(N^3)$ in the tables. Besides, as the values of $3\|\mathbf{v}_i\|^4$, $\forall\, i = 1 \ldots R$ and $\mathbf{M}^{-1}$ depend on the chosen set $(\mathbf{v})_{i=1\ldots R}$ but neither on the sources nor on the mixing process, they can be stored in the working directory and the numbers of operations to compute them are thus not counted.

TABLE I
NUMBER OF FLOATING-POINT OPERATIONS NEEDED BY INITIALIZATION AND OPTIMIZATION STAGES OF FASTICA ALGORITHM

Initialization stage	*Number of operations*
$\mathbf{R_x}$ estimation	$2TN^2$
$\mathbf{R_x}$ eigendecomposition , $\mathbf{B}=\mathbf{\Delta}^{-1/2}\mathbf{E}^t$	$O(N^3)$
$\mathbf{z}(n)=\mathbf{B}\mathbf{x}(n)$	$TN(2N-1)$
One optimization iteration	*Number of operations*
$\forall i \quad \mathbf{w}_i'=E\left\{\mathbf{z}(n)(\mathbf{w}_i^t\mathbf{z}(n))^3\right\}-3\mathbf{w}_i$	$T(4N^2+N)+2N^2$

TABLE II
NUMBER OF FLOATING-POINT OPERATIONS NEEDED BY INITIALIZATION AND OPTIMIZATION STAGES OF O-FICA ALGORITHM

Initialization stage	*Number of operations*
$\mathbf{R_x}$ estimation	$2TN^2$
$\mathbf{R_x}$ eigendecomposition , $\mathbf{B}=\mathbf{\Delta}^{-1/2}\mathbf{E}^t$	$O(N^3)$
$\mathbf{U}=\mathbf{B}^t\mathbf{V}$	$RN(2N-1)$
$\forall i=1,\cdots,R:\ y(n)=\mathbf{u}_i^t\mathbf{x}(n)$	$RT(2N-1)$
$y'(n)=y^4(n)$	$3RT$
$\mu_4=E\{y'(n)\}$	RT
$k(i)=\mu_4-3\|\mathbf{v}_i\|^4$	R
$\boldsymbol{\alpha}=\mathbf{M}^{-1}\mathbf{k}$	$(2R-1)R^2$
One optimization iteration	*Number of operations*
$\forall i,j\ \ w_{ji}'=\sum_{r=1}^{R}\alpha_r d_r(j) w_{ji}^{max(d_r(j)-1,0)}\prod_{k\neq j} w_{ki}^{d_r(k)}$	$(6R-1)N^2$

In [9], we introduced the so-called differential kurtosis that masks the effect of the noise sources and thus allows the partial separation of the nonstationary sources of interest.

DFICA consists of an extension of the kurtosis-based FastICA algorithm to underdetermined mixtures when the sources are divided into two types: $\tilde{N}$ sources of interest, which are long-term nonstationary between two instants n_1 and n_2 with $\tilde{N}\leq P$, and an arbitrary number of long-term stationary noise sources.

Like the standard FastICA algorithm, our DFICA method is composed of two stages: a differential sphering process followed by a fixed-point algorithm. The differential sphering process estimates the differential correlation matrix of the observations defined as the difference between the correlation matrices, respectively, estimated on the two time intervals associated to n_1 and n_2:

$$\mathbf{D}\mathbf{R}_{\mathbf{x}}(n_1,n_2)=\mathbf{R}_{\mathbf{x}}(n_2)-\mathbf{R}_{\mathbf{x}}(n_1). \tag{17}$$

Then, provided $\mathbf{D}\mathbf{R}_{\mathbf{x}}(n_1,n_2)$ is positive definite,[6] its eigendecomposition is computed, which gives an orthogonal matrix $\mathbf{E}$ and a diagonal matrix $\mathbf{\Delta}$. The differential sphering matrix is then obtained as $\mathbf{B}=\mathbf{\Delta}^{-1/2}\mathbf{E}^t$. Then, the differentially sphered observations are computed by using (3).[7] We proved in [9] that the maximization of the absolute value of the differential kurtosis of $\mathbf{w}^t\mathbf{z}(n)$, defined by

$$D\mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n_1,n_2)=\mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n_2)-\mathrm{kurt}_{\mathbf{w}^t\mathbf{z}}(n_1) \tag{18}$$

with respect to $\mathbf{w}$ under the constraint $\|\mathbf{w}\|=1$ achieves the partial separation of the nonstationary sources, i.e., the output signal $y(n)=\mathbf{w}^t\mathbf{z}(n)$ is then composed of a unique long-term nonstationary source superimposed with all stationary sources.

[6]We proved in [9] that $\mathbf{D}\mathbf{R}_{\mathbf{x}}(n_1,n_2)$ is positive definite if and only if the source powers increase between the two times n_1 and n_2; if all these powers decrease, we can permute the two time intervals associated to n_1 and n_2. In order to choose the time intervals associated to n_1 and n_2, we can for instance select some couple of intervals randomly until the necessary condition, i.e., the positive definiteness of the differential correlation matrix, is verified.

[7]If $P>\tilde{N}$, we should use a differential PCA process which consists in using matrices $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{\Delta}}$ instead of $\mathbf{E}$ and $\mathbf{\Delta}$, where $\tilde{\mathbf{E}}$ is composed of the $\tilde{N}$ eigenvectors associated to the highest eigenvalues of $\mathbf{D}\mathbf{R}_{\mathbf{x}}(n_1,n_2)$, and $\tilde{\mathbf{\Delta}}$ is the diagonal matrix composed of these eigenvalues.

The differential kurtosis of $y(n)$, as a difference of kurtoses between two instants, is thus also a multivariate polynomial with respect to the variables $w(1),\ldots,w(\tilde{N})$. We can therefore optimize it with the same principle as in the previous subsection, which first consists in identifying the corresponding polynomial coefficients $(\alpha_r)_{r=1\ldots R}$ and second in applying the same update iterations as DFICA but in the computation space associated to these polynomial coefficients.

III. EXPERIMENTAL RESULTS

In this section, we present experimental results which compare the speeds of our O-FICA and O-DFICA algorithms with the standard FastICA and DFICA methods. In the first series of tests, we compared the computation times of the algorithms until they reach the stopping criterion that is used in the Matlab FastICA Package [12]. To put it briefly, this criterion compares the directions of the extraction vectors after the last two updates, and the optimization is stopped if these directions do not vary more than a certain threshold (we set this threshold parameter to 10^{-4}, which is the default value in the package). We tested the parallel and deflation versions of the algorithms to separate sources with uniform and Laplacian distributions in about equal numbers. For the differential algorithms O-DFICA and DFICA, we took two time intervals between which the powers of the sources of interest vary in the same way. On these intervals, we mixed them with three stationary noise sources corresponding to uniform, Gaussian and Laplacian distributions.

Fig. 1(a) and (b) represents the values of t'/t depending on the number of samples T, where t' and t are, respectively, the computation times of the optimized and standard algorithms. For each configuration, we made 100 Monte Carlo simulations by varying the sources, and the mixing coefficients with a uniform distribution in $[0,1]$, and we averaged the values of t'/t. We can see in Fig. 1(a) and (b) that for a number of sources (a number of nonstationary sources in the differential case) up to 7 (up to 6 for the differential versions) the ratio t'/t is notably lower than 1 when the number T of samples is high enough and decreases when T increases, which shows the effectiveness of our optimized methods. In addition, we compared our O-FICA method with the JADE algorithm which also performs the estimations with all samples once for all before beginning the optimization iterations,

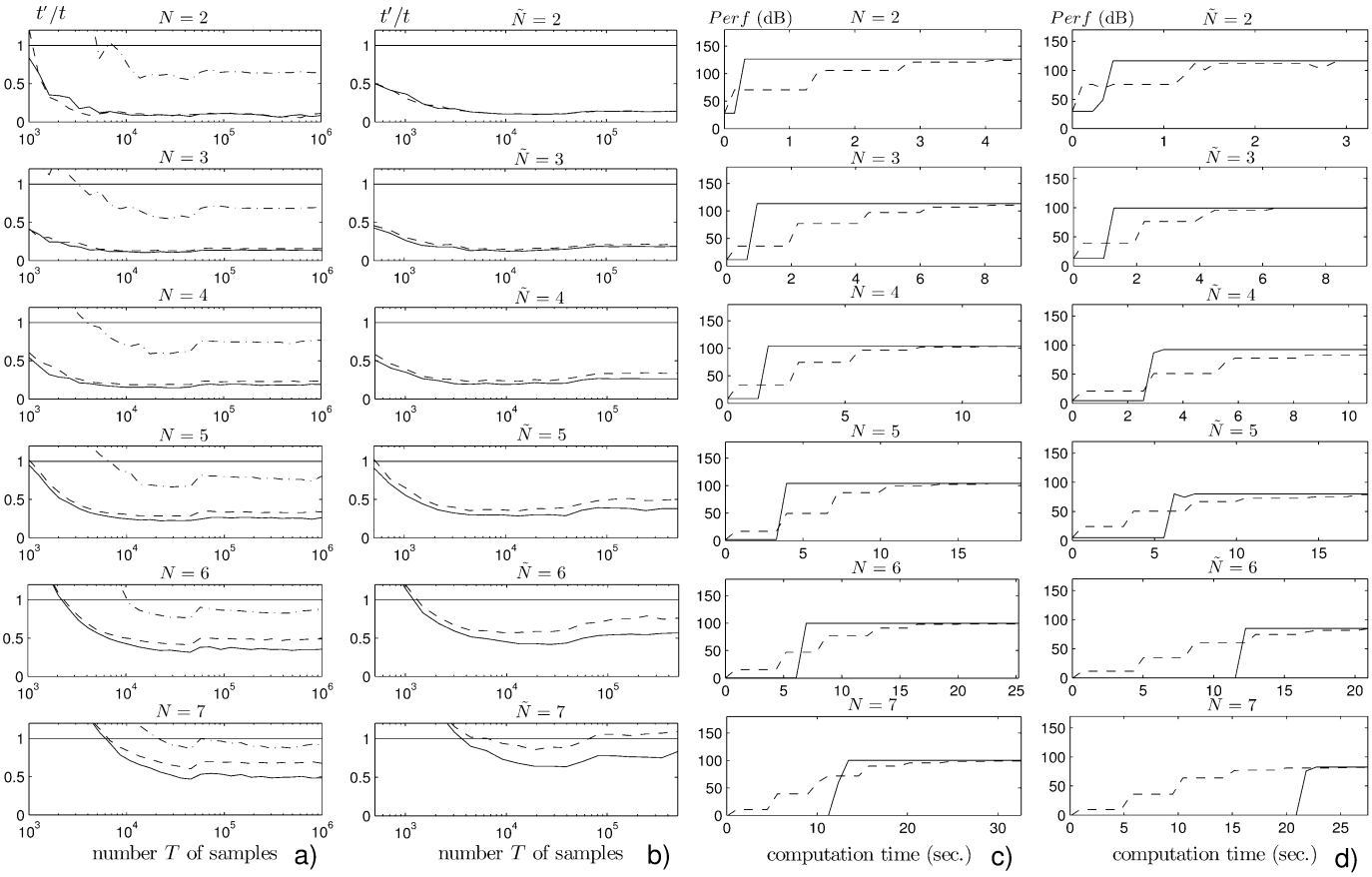

Fig. 1. a) and b): t'/t depending on the number T of samples. a) plain lines: ratio of symmetrical O-FICA (t') and symmetrical FastICA algorithms (t), dashed lines: ratio of deflation O-FICA (t') and deflation FastICA algorithms (t), dot-dashed lines: ratio of symmetrical O-FICA (t') and JADE algorithms (t). b) plain lines: ratio of symmetrical O-DFICA (t') and DFICA algorithms (t), dashed lines: ratio of deflation O-DFICA (t') and DFICA algorithms (t). c) and d): Performance index $Perf$ depending on the computation time, plain lines: optimized algorithms, dashed lines: standard algorithms. c) (over)determined FastICA and O-FICA algorithms, d) underdetermined DFICA and O-DFICA algorithms.

but uses another criterion, based on the cancellation of fourth-order cross-cumulants. Our tests also show that our method yields somewhat better performance than the JADE algorithm.

In the second series of experiments, we represented the performance index depending on the allowed computation time for the symmetrical standard and optimized algorithms. The above-mentioned performance index is defined as

$$\mathrm{Perf} = \mathrm{mean}_j\left(\max_i 10\ \log_{10}\left(g_{ij}^2/\sum_{j\neq i} g_{ij}^2\right)\right) + \mathrm{mean}_i\left(\max_j 10\ \log_{10}\left(g_{ij}^2/\sum_{j\neq i} g_{ij}^2\right)\right)$$

where the g_{ij} values are the entries of the performance matrix $\mathbf{G}$, which is defined as the product of the mixing and separating matrices. In the differential case, we only take into account the columns of $\mathbf{G}$ associated with the nonstationary sources and thus evaluate the quality of the partial separation of the sources of interest. As previously, for each configuration associated with a number of sources and an allowed processing time, we made 100 Monte Carlo simulations by varying the sources (with uniform and Laplacian distributions in about equal numbers) and the mixing coefficients (with a uniform distribution in $[0, 1]$) and then averaged the values of Perf. We chose an overall number of samples equal to 100 000 ($T = 50\,000$ in the differential case). Fig. 1(c) and (d) show that the optimized algorithms reach the highest value of Perf with a significantly shorter processing time (except for $\tilde{N} = 7$ in the differential case). We also see that for the standard FastICA and DFICA algorithms, the performance index progressively increases, contrary to our optimized versions for which the index increases suddenly at the time when the polynomial parameters (α_r) have been estimated (the period when the performance criterion is low corresponds to the polynomial identification stage), as the optimization updates are then performed in a very efficient computation space.

IV. Conclusion

In this paper, we have presented a new method for optimizing the kurtosis-based separation criteria used by the well-known FastICA algorithm and by our recent DFICA algorithm intended for underdetermined partial separation of nonstationary sources. We proved that these criteria can be expressed as multivariate polynomials and proposed a fast method to identify their coefficients. This lets us realize their constrained optimization in a simpler computation space as compared to the space associated to the classical FastICA and DFICA algorithms. Our approach is particularly attractive in terms of computation time and requested memory space when the observations contain a large number of samples, which is often encountered in blind image separation for instance.

$$E_1 = \left\{ \mathbf{w} \in \{0,1\}^N \setminus \exists i_1, w(i_1) = 1, \forall\, i \neq i_1 \;\; w(i) = 0 \right\}$$
$$E_2 = \left\{ \mathbf{w} \in \{0,1\}^N \setminus \exists i_1, i_2, w(i_1) = w(i_2) = 1, \forall\, i \neq i_1, i_2 \;\; w(i) = 0 \right\}$$
$$E_3 = \left\{ \mathbf{w} \in \{0,1,2\}^N \setminus \exists i_1, i_2, w(i_1) = 2w(i_2) = 2, \forall\, i \neq i_1, i_2 \;\; w(i) = 0 \right\}$$
$$E_4 = \left\{ \mathbf{w} \in \{0,1,2\}^N \setminus \exists i_1, i_2, i_3, w(i_1) = 2w(i_2) = 2w(i_3) = 2, \forall\, i \neq i_1, i_2, i_3 \;\; w(i) = 0 \right\}$$
$$E_5 = \left\{ \mathbf{w} \in \{0,1\}^N \setminus \exists i_1, \ldots, i_4, w(i_1) = \cdots = w(i_4) = 1, \forall\, i \neq i_1, \ldots, i_4 \;\; w(i) = 0 \right\}.$$

APPENDIX
PROPOSED FAMILY OF EXTRACTION VECTORS

Let us define the five sets[8]: see the equation shown at the top of the page.

The cardinalities of $E_1, \ldots, E_5$ are, respectively, $N, N(N-1)/2$, $N(N-1)$, $N(N-1)(N-2)/2$, $N(N-1)(N-2)(N-3)/24$. By denoting $E = \cup_{i=1}^{5} E_i$, we have $\mathrm{card}(E) = \sum_{i=1}^{5} \mathrm{card}(\mathbf{E}_i) = N + 3N(N-1)/2 + N(N-1)(N-2)/2 + N(N-1)(N-2)(N-3)/24 = \mathrm{card}(D) = R$. We numerically verified that using this family as the vectors $\mathbf{v}_i$ gives a nonsingular matrix $\mathbf{M}$ as defined in (13) for a number of sources $N \leq 7$. The respective conditioning numbers of $\mathbf{M}$ (defined as the ratio of the greatest and lowest eigenvalues of $\mathbf{M}$) for $N = 2, \ldots, 7$ are indeed $182, 414, 844, 1605, 2758, 4344$ which is very low given the associated values of R (respectively, $5, 15, 35, 70, 126, 210$). For instance, the mean conditioning number of 210-dimensional matrices with coefficients uniformly distributed between 0 and 1 is greater than 50 000. This family of vectors $(\mathbf{v}_i)_{i=1..R}$ may then be used to identify the set of coefficients $(\alpha_r)_{r=1..R}$ as defined in (10).

To illustrate this choice of family, let us describe each set $E_i, i \in 1 \ldots 5$ in the simplest cases, i.e., $N = 2$ and $N = 3$. For $N = 2$, we have $E_1 = \{(1,0);(0,1)\}$, $E_2 = \{(1,1)\}$, $E_3 = \{(2,1);(1,2)\}$, $E_4 = \emptyset$, $E_5 = \emptyset$. For $N = 3$, $E_1 = \{(1,0,0);(0,1,0);(0,0,1)\}$, $E_2 = \{(1,1,0);(1,0,1)(0,1,1)\}$, $E_3 = \{(2,1,0);(1,2,0);(2,0,1);(1,0,2);(0,2,1);(0,1,2)\}$, $E_4 = \{(2,1,1);(1,2,1);(1,1,2)\}$, $E_5 = \emptyset$.

REFERENCES

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[2] P. Comon, "Independent component analysis," in *Proc. Int. Signal Processing Workshop Higher-Order Statistics (HOS'91)*, Chamrousse, France, Jul. 1991, pp. 111–120.

[3] P. Comon, "Independent component analysis, a new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[4] E. Moreau and O. Macchi, "New self-adaptive algorithms for source separation based on contrast functions," in *Proc. Workshop Higher-Order Statistics (HOS'93)*, Lake Tahoe, NV, Jun. 1993, pp. 215–219.

[5] P. Comon and E. Moreau, "Improved contrast dedicated to blind separation in communications," in *Proc. Intl. Conf. Acoustics, Speech, Signal Processing (ICASSP'97)*, Munich, Germany, Apr. 1997, pp. 3453–3456.

[6] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A deflation approach," *Signal Process.*, vol. 45, no. 1, pp. 59–83, Jul. 1995.

[7] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, Oct. 1997.

[8] M. Funaro, E. Oja, and H. Valpola, "Independent component analysis for artefact separation in astrophysical images," *Neural Netw.*, vol. 16, pp. 469–478, 2003.

[8]In the underdetermined case, N must be replaced by $\bar{N}$ in the following.

[9] J. Thomas, Y. Deville, and S. Hosseini, "Differential fast fixed-point algorithms for underdetermined instantaneous and convolutive partial blind source separation," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3717–3729, Jul. 2007.

[10] V. Zarzoso, P. Comon, and M. Kallel, "How fast is FastICA?," in *Proc. Eur. Signal Processing Conf. (EUSIPCO'06)*, Florence, Italy, Sep. 2006.

[11] E. Oja and Z. Yuan, "The FastICA algorithm revisited: Convergence analysis," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1370–1381, Nov. 2006.

[12] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen, FastICA for Matlab 7.x and 6.x, Version 2.5 2005 [Online]. Available: http://www.cis.hut.Fi/projects/ica/fastica/code/dlcode.shtml

[13] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, May 1999.

On the TT-Transform and Its Diagonal Elements

Carine Simon, Martin Schimmel, and Juanjo José Dañobeitia

***Abstract*—The TT-transform stands for time–time transform and has been derived as an inverse Fourier transform of the time-frequency S-transform. Up to date, only the diagonal of the TT-transform has been used for signal characterization. We show here an alternative and simplified derivation of the TT-transform which enables a better understanding of this transform. In particular, we demonstrate that the diagonal elements of the TT-transform represent a simple frequency filtered version of the original signal and, thus, that little additional information is gained through the TT-transform.**

***Index Terms*—Local spectra, S-transform, time-frequency localization, time–time analysis, time-varying filters, TT-transform.**

I. INTRODUCTION

IN disciplines such as music or geophysics, signals are nonstationary. The need for processing such signals has led to the appearance of several types of time varying frequency filters, such as the short time Fourier transform [1], wavelets [2], and more recently the

Manuscript received November 22, 2007; revised June 26, 2008. First published August 19, 2008; current version published October 15, 2008. The associate editor coordinating the review of this paper and approving it for publication was Prof. Brian L. Evans. This work was supported by the project SigSensual ref. REN2003-08341-C03- C01-02 and CTM2004-04510-C03-02. The work of M. Schimmel is supported through the Ramon y Cajal and the Consolider-Ingenio 2010 Nr. CSD2006-00041 program.

C. Simon and J. J. Dañobeitia are with the Marine Technology Unit, CSIC, 08003 Barcelona, Spain (e-mail: csimon@cmima.csic.es; jjdanobeitia@cmima.csic.es).

M. Schimmel is with the Institute of Earth Sciences Jaume Almera, CSIC, 08028 Barcelona, Spain (e-mail: schimmel@ija.csic.es).

Digital Object Identifier 10.1109/TSP.2008.929670