ECMS 2007 & Doctoral School
8th International workshop on Electronics, Control, Modelling, Measurement and Signals 2007
& Doctoral School (EDSYS, GEET), May 21-23, 2007, Liberec, Czech Republic
----

Multivariate polynomial identification for blind image separation

J. Thomas[1], Y. Deville[1], S. Hosseini[1]
[1] Université Paul Sabatier Toulouse 3 - CNRS, Laboratoire d'Astrophysique de Toulouse Tarbes,
14 av. Edouard Belin, Toulouse, France

jthomas,ydeville,shosseini@ast.obs-mip.fr

Abstract: *This paper presents a new approach for optimizing the non-normalized kurtosis under unit power constraint. It is inspired from the famous FastICA algorithm which uses a fixed-point algorithm and computes fourth-order statistics at each step of the optimization, which is very demanding in terms of both computation time and memory space, especially when the number of samples is high. Our method avoids this by using a polynomial identification at the beginning of the algorithm, which lets us perform the optimization in a more efficient computation space. Our so-called O-FICA algorithm is particularly interesting in blind image separation because of the present size increase of light sensors.*

I. INTRODUCTION

This paper deals with fast blind separation of linear instantaneous mixtures [1] when the mixture recordings contain a high number of samples. This configuration is frequent, especially in image separation because of the great number of pixels in CCD and CMOS-type light sensors. We here present a fixed-point algorithm inspired from FastICA which avoids the computation of statistics at each step of the optimization algorithm of the kurtosis, thanks to a first step based on polynomial identification.
A linear instantaneous mixture can be written in matrix form as

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) \tag{1}$$

where $\mathbf{A}$ is an N×N mixing matrix and $\mathbf{s}(n)$ and $\mathbf{x}(n)$ are respectively the source and observation N-size vectors. Besides the classical assumptions used by Independent Component Analysis (independence and non-Gaussianity of the sources, full-column rank matrix $\mathbf{A}$), we here suppose that the observations contain a great number of samples (typically more than 100000, which is often the case for images). We also suppose that the number of sources is not too high (up to 7 typically), which is often the case except in some biomedical applications which imply a great number of recorded channels.

II. THE FASTICA ALGORITHM

Let us summarize the kurtosis-based FastICA algorithm from which our method is inspired. It is composed of a first step, called sphering, which is realized by the following operation

$$\mathbf{z}(n) = \mathbf{R}_{\mathbf{x}}^{-1/2}\mathbf{x}(n) \tag{2}$$

where $\mathbf{R}_{\mathbf{x}}$ is the correlation matrix of the observation vector and $\mathbf{z}(n)$ is the sphered observation vector which meets the condition $\mathbf{R}_{\mathbf{z}} = \mathbf{I}$.

After this sphering process, the FastICA algorithm maximizes the absolute value of the kurtosis of $\mathbf{w}^t\mathbf{z}(n)$, $kurt(\mathbf{w}^t\mathbf{z}(n))$, with respect to an extraction vector $\mathbf{w}$ under the constraint $\|\mathbf{w}\| = 1$. To this end, the parallel version of FastICA developed by Hyvärinen and Oja [2] adapts at the same time all the extraction vectors $\mathbf{w}_i$ associated with the N sources, thus providing output signals $y_i(n) = \mathbf{w}_i^t\mathbf{z}(n)$, and orthogonalizes these vectors

$\mathbf{w}_i$ to prevent them from converging towards identical points (up to sign). This orthogonalization can be made by the symmetrical operation $\mathbf{W} = \mathbf{W}(\mathbf{W}^t\mathbf{W})^{-1/2}$ where $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_N]$ contains the N extraction vectors arranged column-wise.

The symmetrical FastICA algorithm can then be summarized by the following procedure:

- Sphere the observation vector by using (2).
- Initialize $\mathbf{W}$ with N vectors of size N arranged column-wise (one can choose $\mathbf{W} = \mathbf{I}_N$).
- Repeat the following steps 1) and 2) until convergence:

$$\begin{aligned} &1)\ \forall k = 1...N,\ \mathbf{w}_k \leftarrow E\left\{\mathbf{z}(n)\left(\mathbf{w}_k^t\mathbf{z}(n)\right)^3\right\} - 3\mathbf{w}_k \propto \frac{\partial kurt(\mathbf{w}_k^t\mathbf{z}(n))}{\partial \mathbf{w}_k} \\ &2)\ \mathbf{W} \leftarrow \mathbf{W}\left(\mathbf{W}^t\mathbf{W}\right)^{-1/2} \end{aligned} \tag{3}$$

This algorithm is very popular in the BSS community because it is fast and has no tunable parameter. Besides, its convergence has been rigorously proved by Oja and Yuan [3]. This explains its great success in many domains, such as astrophysical image separation for instance [4].

Nevertheless, we can note that one iteration of the fixed-point algorithm is very consuming in terms of computation time and requested memory space, especially when the number of samples T is high (this is often the case in astronomy for instance). Indeed, in the FastICA Matlab package [5], the fixed-point update is realized by the following matrix operation

W = (Z*((Z'*W).^3)) / T - 3*W (4)

where Z is an N×T-size matrix containing the T samples of the sphered observations and W is an N×N matrix containing the extraction vectors. As operation (4) requests to store Z and (Z'*W).^3, it needs 2TN memory words. Furthermore one can show that (4) needs the computation of $T(4N^2+N)+2N^2$ elementary operations (additions and multiplications), which is very demanding if the number of samples T is high.

III. PROPOSED METHOD

We here propose a method which avoids this important request in memory and computation during the kurtosis optimization. Our method is based on a multivariate polynomial identification which makes it possible to perform the fixed-point optimization in a simpler computation space. We also avoid the computation and the storage of the sphered observation vector $\mathbf{z}(n)$ defined by (2).

Let us denote by $y(n) = \mathbf{w}^t\mathbf{z}(n)$ a linear combination of the sphered observations in $\mathbf{z}(n)$ computed as in the FastICA algorithm (the computation of $\mathbf{z}(n)$ will be avoided in our approach as explained further). We suppose that all the signals are centered, so that the non-normalized kurtosis of $y(n)$ reads:

$$\begin{aligned} kurt(y(n)) &= kurt(\mathbf{w}^t\mathbf{z}(n)) \\ &= E\left\{\left(\mathbf{w}^t\mathbf{z}(n)\right)^4\right\} - 3E\left\{\left(\mathbf{w}^t\mathbf{z}(n)\right)^2\right\}^2. \end{aligned} \tag{5}$$

As we have

$$E\left\{\left(\mathbf{w}^t\mathbf{z}(n)\right)^2\right\} = E\left\{\mathbf{w}^t\mathbf{z}(n)\mathbf{z}^t(n)\mathbf{w}\right\} = \mathbf{w}^t E\left\{\mathbf{z}(n)\mathbf{z}^t(n)\right\}\mathbf{w} = \mathbf{w}^t\mathbf{I}\mathbf{w} = \|\mathbf{w}\|^2, \tag{6}$$

(5) becomes

$$
\begin{aligned}
kurt(y(n)) &= E\left\{ \sum_{i_1,i_2,i_3,i_4=1}^{N} \prod_{k=1}^{4} w(i_k) z_{i_k}(n) \right\} - 3\left(\sum_{i=1}^{N} w(i)^2 \right)^2 \\
&= \sum_{i_1,i_2,i_3,i_4=1}^{N} E\left\{ \prod_{k=1}^{4} z_{i_k}(n) \right\} \left(\prod_{k=1}^{4} w(i_k) \right) - 3 \sum_{i_1,i_2=1}^{N} w(i_1)^2 w(i_2)^2 .
\end{aligned} \tag{7}
$$

Since the result of (7) is a fourth-order polynomial of the variables $w(1),\ldots,w(N)$, we can define a set of coefficients $(\alpha_{\mathbf{d}})_{\mathbf{d}\in D}$ with $D = \left\{ \mathbf{d} \in \{0,\ldots,4\}^N \setminus \sum_{k=1}^{N} d(k) = 4 \right\}$ so that

$$
kurt(y(n)) = \sum_{\mathbf{d}\in D} \alpha_{\mathbf{d}} \prod_{k=1}^{N} w(k)^{d(k)} \tag{8}
$$

which corresponds to the canonical development of the non-normalized kurtosis. Let us denote by R the cardinality of D, its elements as $\mathbf{d}_1,\ldots,\mathbf{d}_R$ and the values of $(\alpha_{\mathbf{d}})_{\mathbf{d}\in D}$ as $\alpha_1,\ldots,\alpha_R$.
We have then

$$
kurt(y(n)) = \sum_{r=1}^{R} \alpha_r \prod_{k=1}^{N} w(k)^{d_r(k)} . \tag{9}
$$

It may be shown that the cardinality R of D is equal to $N + \frac{3N(N-1)}{2} + \frac{N(N-1)(N-2)}{2} + \frac{N(N-1)(N-2)(N-3)}{24}$.
It would be possible to compute directly 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 more efficient method based on the following observation. Let us denote by $(\mathbf{v}_i)_{i=1..R}$ a family of R vectors of size N. Replacing $\mathbf{w}$ by $\mathbf{v}_i$ in (9), we have

$$
\forall i = 1\ldots R, \quad kurt(\mathbf{v}_i^t \mathbf{z}(n)) = \sum_{j=1}^{R} \alpha_j \prod_{k=1}^{N} v_i(k)^{d_j(k)} \tag{10}
$$

which can be written as

$$
\mathbf{M}\boldsymbol{\alpha} = \mathbf{k} \tag{11}
$$

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

$$
\begin{cases}
\forall i, j = 1\ldots R, & m_{ij} = \prod_{k=1}^{N} v_i(k)^{d_j(k)} \\
\forall i = 1\ldots R, & k_i = kurt(\mathbf{v}_i^t \mathbf{z}(n))
\end{cases} \tag{12}
$$

By choosing R extraction vectors $\mathbf{v}_i$ of size N which imply a non-singular matrix $\mathbf{M}$ and by computing the kurtoses k_i of the R signals $\mathbf{v}_i^t \mathbf{z}(n)$, we can thus identify the R coefficients $(\alpha_r)_{r=1..R}$ with the inverse of (11), i.e. $\boldsymbol{\alpha} = \mathbf{M}^{-1}\mathbf{k}$. Thanks to the properties of the sphered vector $\mathbf{z}(n)$ which imply $E\{(\mathbf{v}_i^t \mathbf{z}(n))^2\} = \|\mathbf{v}_i\|^2$ as proved by (6), the kurtosis of $\mathbf{v}_i^t \mathbf{z}(n)$ is equal to $kurt(\mathbf{v}_i^t \mathbf{z}(n)) = E\{(\mathbf{v}_i^t \mathbf{z}(n))^4\} - 3\|\mathbf{v}_i\|^4$ so that we only need to estimate one fourth-order auto-moment for each of the R extraction vectors. Moreover, the relation

$$\mathbf{v}_i^t\mathbf{z}(n)=\mathbf{v}_i^t\mathbf{R}_\mathbf{x}^{-1/2}\mathbf{x}(n)=\mathbf{u}_i^t\mathbf{x}(n) \tag{13}$$

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

Let us notice that $\mathbf{M}$ is independent from the sources and from the mixing process. One can thus compute its inverse once for all for each considered value of N (up to 7 in our case) and store them. We propose in the Appendix to choose some particular vectors $(\mathbf{v}_i)_{i=1..R}$ which yield a well-conditioned matrix $\mathbf{M}$.

After we have identified the set of coefficients $(\alpha_\mathrm{r})_{\mathrm{r}=1..\mathrm{R}}$, we want to maximize the absolute value of $kurt(\mathbf{w}^t\mathbf{z}(n))=\sum_{r=1}^{R}\alpha_r\prod_{k=1}^{N}w(k)^{d_r(k)}$ with respect to $\mathbf{w}$ under the constraint $\|\mathbf{w}\|=1$, as in the FastICA criterion. This is here realized in a similar way as in the FastICA algorithm, by means of a fixed-point algorithm, but this time in a different computation space. The gradient of $kurt(\mathbf{w}^t\mathbf{z}(n))$ with respect to $\mathbf{w}$ in our space reads

$$\frac{\partial kurt(\mathbf{w}^t\mathbf{z}(n))}{\partial\mathbf{w}}=\sum_{r=1}^{R}\alpha_r\frac{\partial\prod_{k=1}^{N}w(k)^{d_r(k)}}{\partial\mathbf{w}} \tag{14}$$

with

$$\frac{\partial\prod_{k=1}^{N}w(k)^{d_r(k)}}{\partial w(j)}=\begin{cases}d_r(j)w(j)^{d_r(j)-1}\prod_{k\neq j}w(k)^{d_r(k)} & \text{, if } d_r(j)>0\\ 0 & \text{, if } d_r(j)=0\end{cases} \tag{15}$$

Let us now summarize our parallel fixed-point algorithm, optimized for long mixture recordings, that we call O-FICA:

- Estimate the correlation matrix $\mathbf{R}_\mathbf{x}$ of the observation vector $\mathbf{x}(n)$.
- Compute $\mathbf{U}=\mathbf{R}_\mathbf{x}^{-1/2}\mathbf{V}$ where $\mathbf{V}$ is composed of R extraction vectors arranged column-wise (we propose a particular set of vectors in the Appendix).
- Compute the kurtoses $(\mathrm{k}_\mathrm{i})_{\mathrm{i}=1..\mathrm{R}}$ of the R signals $\mathbf{u}_i^t\mathbf{x}(n),\ i=1...R$ and then determine the set of coefficients $\boldsymbol{\alpha}=(\alpha_r)_{r=1..R}$ thanks to the relation $\boldsymbol{\alpha}=\mathbf{M}^{-1}\mathbf{k}$ (by means of the pre-computed inverse of $\mathbf{M}$).
- Initialize $\mathbf{W}$ with N independent N-size vectors arranged column-wise and then repeat 1) and 2) below until convergence:

$$1)\quad \forall i,j=1...N,\quad w_{ji}\leftarrow\sum_{r=1}^{R}\alpha_r d_r(j)w_{ji}^{\max(d_r(j)-1,0)}\prod_{k\neq j}w_{ki}^{d_r(k)} \tag{16}$$

which is equivalent to setting $\forall i=1...N,\ \ \mathbf{w}_i\leftarrow\frac{\partial kurt(\mathbf{w}_i^t\mathbf{z}(n))}{\partial\mathbf{w}_i}$.

$$2)\quad \mathbf{W}\leftarrow\mathbf{W}(\mathbf{W}^t\mathbf{W})^{-1/2} \tag{17}$$

Contrary to the FastICA algorithm, our method does not compute statistics at each iteration of the fixed-point optimization. Here, we only manipulate some polynomial coefficients α_r, which is faster for a large value of T, as their number is independent from T (these coefficients must not be too numerous yet, which is the case for $N \leq 7$). Furthermore, we avoid to store Z and (Z'*W).^3 which represent 2TN memory words in the FastICA algorithm. In our case, it may be shown that we only use $R+RN+R^2+N^2$ memory words, which is independent from the number of samples T.
Our algorithm doing the same update operation as FastICA (in another computation space), it reaches at each iteration the same point for each of the extraction vectors and thus gives the same separation performance with a smaller processing time.

IV. EXPERIMENTAL RESULTS

In this section, we present experimental results which compare the speeds of our O-FICA and of the standard FastICA algorithm. We measured the computation times of the algorithms until they reach the stopping criterion that is used in the Matlab FastICA package. To put it briefly, this criterion compares the directions of the extraction vectors after the last two updates and the optimization stops if these directions do not vary more than a certain threshold (we took a threshold parameter of 10^{-4} which is the default value in the package).

Fig. 1 represents the mean value of t'/t depending on the number of samples T, with t' and t being respectively the computation times of the O-FICA and FastICA algorithms. For each configuration, we made 100 Monte-Carlo simulations and averaged the values of t'/t. We can see on Fig. 1 that for a number of sources up to 7, the ratio t'/t is notably lower than 1 when the number of samples T is high enough and decreases (down to 1/15 in some cases) when T increases.

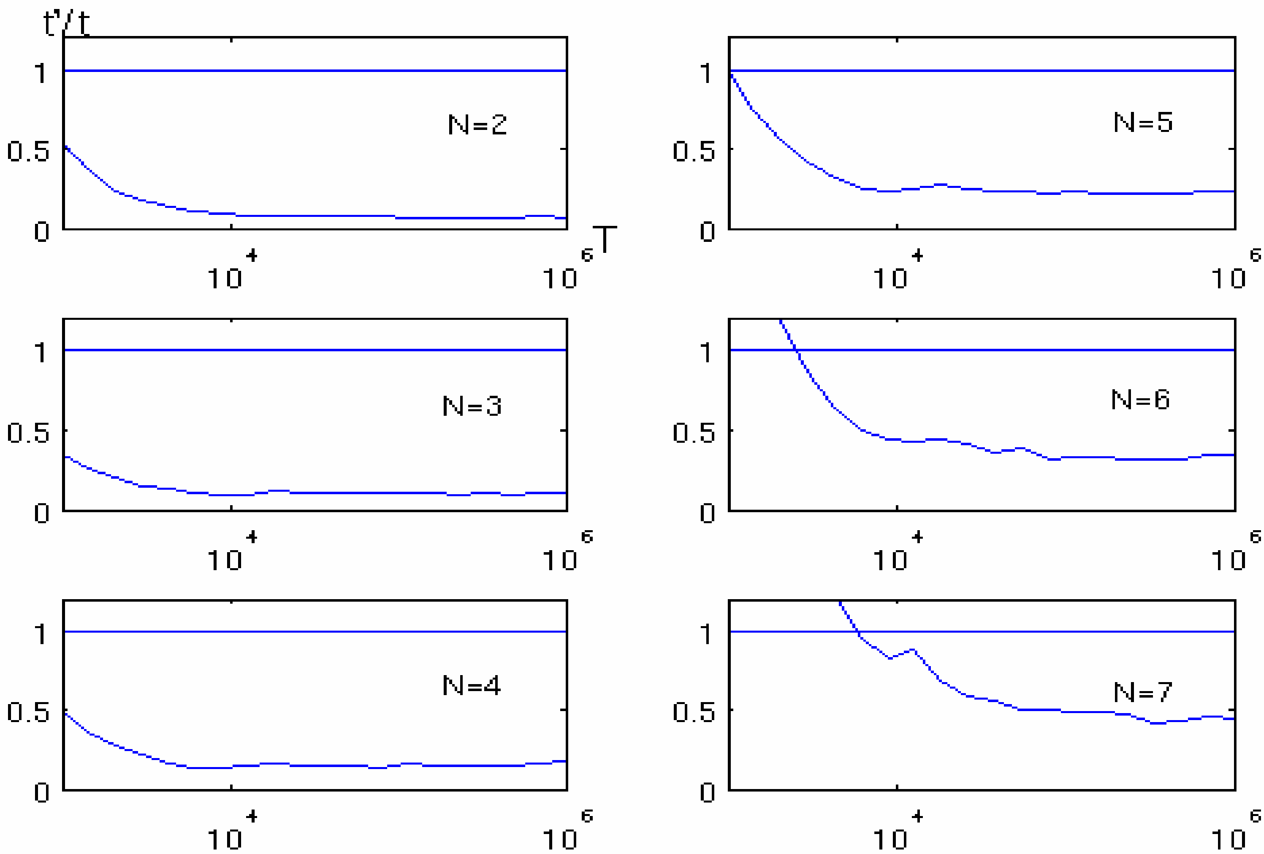

Fig. 1: Ratio of processing times depending on the number T of samples for various numbers N of sources.

V. CONCLUSION

We have presented a new method for the optimization of the non-normalized kurtosis under unit power constraint. We base our approach on a polynomial identification step that allows us to use a fixed-point algorithm in a more efficient computation space. Our O-FICA algorithm is especially interesting when the recorded signals contain a high number of samples, which is often the case in blind image separation. Compared to the FastICA algorithm, our algorithm is up to 15 times faster for certain configurations.

APPENDIX: CHOICE OF THE R-SIZE FAMILY OF EXTRACTION VECTORS

Let us define the five sets:

$$\begin{aligned}
E_1 &= \left\{ \mathbf{w} \in \{0,1\}^N \;\backslash\; \exists i_1 \;,\; w(i_1)=1,\;\; \forall i \neq i_1\; w(i)=0 \right\} \\
E_2 &= \left\{ \mathbf{w} \in \{0,1\}^N \;\backslash\; \exists i_1, i_2 \;,\; w(i_1)=w(i_2)=1,\;\; \forall i \neq i_1, i_2\; w(i)=0 \right\} \\
E_3 &= \left\{ \mathbf{w} \in \{0,1,2\}^N \;\backslash\; \exists i_1, i_2 \;,\; w(i_1)=2w(i_2)=2,\;\; \forall i \neq i_1, i_2\; w(i)=0 \right\} \\
E_4 &= \left\{ \mathbf{w} \in \{0,1,2\}^N \;\backslash\; \exists i_1, i_2, i_3 \;,\; w(i_1)=2w(i_2)=2w(i_3)=2,\;\; \forall i \neq i_1, i_2, i_3\; w(i)=0 \right\} \\
E_5 &= \left\{ \mathbf{w} \in \{0,1\}^N \;\backslash\; \exists i_1, i_2, i_3, i_4 \;,\; w(i_1)=\ldots=w(i_4)=1,\;\; \forall i \neq i_1, \ldots, i_4\; w(i)=0 \right\}.
\end{aligned} \quad (18)$$

The cardinalities of $E_1,\ldots,E_5$ are $Card(E_1) = N$, $Card(E_2) = \frac{N(N-1)}{2}$, $Card(E_3) = N(N-1)$, $Card(E_4) = \frac{N(N-1)(N-2)}{2}$, $Card(E_5) = \frac{N(N-1)(N-2)(N-3)}{24}$. Then by denoting $E = \bigcup_{i=1}^{5} E_i$,

$$Card(E) = \sum_{i=1}^{5} Card(E_i) = N + \frac{N(N-1)}{2} + N(N-1) + \frac{N(N-1)(N-2)}{2} + \frac{N(N-1)(N-2)(N-3)}{24} = Card(D) = R.$$

We numerically verified that using this family as the vectors $\mathbf{v}_i$ gives a non-singular matrix $\mathbf{M}$ as defined in (12) 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 rather 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 50000. Then we have a family of vectors $\mathbf{v}_i$ that may be used to identify the set of coefficients $(\alpha_r)_{r=1..R}$ as defined by (9).

REFERENCES

[1] A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis,* John Wiley & Sons, 2001.
[2] A. Hyvärinen and E. Oja, *A Fast Fixed-Point Algorithm for Independent Component Analysis*, Neural Computation, vol. 9, pp. 1483-1492, Oct. 1997.
[3] E. Oja and Z. Yuan, *The FastICA Algorithm Revisited: Convergence Analysis*, IEEE Trans. on Neural Networks, vol. 17, pp. 1370-1381, Nov. 2006.
[4] M. Funaro, E. Oja and H. Valpola, *Independent component analysis for artefact separation in astrophysical images*, Neural Networks, vol. 16, pp. 469-478, 2003.
[5] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen, *FastICA for Matlab 7.x and 6.x*, Version 2.5, Oct. 19 2005. http://www.cis.hut.fi/projects/ica/fastica.