Blind identification of underdetermined mixtures of complex sources based on the characteristic function

Xavier Luciani, André De Almeida, Pierre Comon

To cite this version:


HAL Id: hal-00512805
https://hal.archives-ouvertes.fr/hal-00512805
Submitted on 31 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
BLIND IDENTIFICATION OF UNDERDETERMINED MIXTURES OF COMPLEX SOURCES BASED ON THE CHARACTERISTIC FUNCTION

Xavier Luciani(1), André L. F. de Almeida(2), Pierre Comon(1)

(1) I3S Laboratory, University of Nice-Sophia Antipolis (UNS), CNRS, France.
Federal University of Ceará, Fortaleza, Brazil.
E-mails: {luciani, pcomon}@i3s.unice.fr, andre@gtel.ufc.br

ABSTRACT

In this work we consider the problem of blind identification of underdetermined mixtures using the generating function of the observations. This approach had been successfully applied on real sources but had not been extended to the more attractive case of complex mixtures of complex sources. This is the main goal of the present study. By developing the core equation in the complex case, we arrive at a particular tensor stowage which involves an original tensor decomposition. Exploiting this decomposition, an algorithm is proposed to blindly estimate the mixing matrix. Three versions of this algorithm based on 2nd, 3rd and 4th-order derivatives of the generating function are evaluated on complex mixtures of 4-QAM and 8-PSK sources and compared to the 6-BIOME algorithm by means of simulation results.

1. INTRODUCTION

Blind Identification (BI) methods have been successfully applied in various scientific areas, including for instance telecommunications [1], acoustic [2] or biomedical signal processing [3]. A large family of BI methods relies on the theory of Independent Component Analysis (ICA) [4] and thereby involves second or higher-order statistics. BI of underdetermined mixtures (when the number of sources exceeds the number of sensors) is an important subcategory of BI problems which arises in many practical situations, especially in telecommunications.

Several solutions have been proposed in the literature to solve this problem (see, e.g. [5], [6], [7], [8], [9]). Notably, some original methods, which do not exploit cumulants but the generating function are available for most applications involving BI. Computer results obtained from simulated telecommunications signals are presented in the last part of the paper as an application example.

2. THEORY

2.1 Notations

Vectors, matrices and tensors are denoted by lower case boldface ($\mathbf{a}$), upper case boldface ($\mathbf{A}$) and upper case calligraphic ($\mathcal{A}$) letters respectively. $a_i$ is the $i$th coordinate of vector $\mathbf{a}$ and $a_{ij}$ is the $i,j$th element of matrix $\mathbf{A}$. The $(i,j)$ entry of matrix $\mathbf{A}$ is denoted $A_{ij}$ and the $(i,j,k)$ entry of the third order tensor $\mathcal{A}$ is denoted $\mathcal{A}_{ijk}$. Complex objects are underlined, their real and imaginary parts are denoted $\Re\{\cdot\}$ and $\Im\{\cdot\}$ respectively. $E[\cdot]$ denotes the mean value of a random variable.

2.2 Blind Identification Problem

Consider a noisy mixture of $K$ statistically independent narrowband sources received by an array of $N$ sensors. The vector $y(m)$ containing discrete observations of the received signal at the sensor outputs is modelled according to the following linear model:

$$ y(m) = Hs(m) + n(m), \quad m = 1 \cdots M $$

Note that in the following we refer to the real or complex case when both mixture and sources belong to $\Re$ ("real case") or $\mathbb{C}$ ("complex case").
where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{C}^{N \times K}$, $\mathbf{s} = [s_1, \ldots, s_n]^T \in \mathbb{C}^K$ and $\mathbf{n} \in \mathbb{C}^N$ are the mixing matrix, source and Gaussian noise random vectors, respectively. It is assumed that for any fixed sample index $m$, $\mathbf{s}$ and $\mathbf{n}$ are statistically independent. The problem is to identify the mixing matrix $\mathbf{H}$ (up to trivial column permutation and scaling) from only knowledge of the observation vector $\mathbf{y}(m)$, using its characteristic function. Recall that we are interested in the so-called underdetermined case, which means that we have $K > N$.

Before to proceed, we describe our working hypotheses:

**H1**: The mixing matrix $\mathbf{H}$ does not contain collinear columns.

**H2**: The sources $\mathbf{s}_1(m), \ldots, \mathbf{s}_n(m)$ are mutually independent and non-Gaussian.

**H3**: The number of sources is known.

The theoretical justification of the present approach is similar to that of the real case. It consists in successively differentiating the second generating function of the observations at different points of the observation space. By working with complex mixtures of complex sources, this leads to a new core equation following a particular tensor decomposition. By exploiting the structure of this tensor decomposition, the mixing matrix is estimated.

### 2.3 The new core equation

The first step is to obtain the new core equation. This is achieved by decomposing the second generating function of the observations as a sum of the individual second generating functions of the sources. Generating functions of a complex variable are actually defined by assimilating $\mathbb{C}$ to $\mathbb{R}^2$. Thus the second generating function of the $k$th source $\mathbf{s}_k$ taken at the point $\mathbf{z}$ of $\mathbb{C}$ is defined as a function of two real variables (real and imaginary parts):

$$
\phi_k(\mathbf{R}(\mathbf{z}), \mathbb{Z}(\mathbf{z})) \overset{\text{def}}{=} \log E[\exp(\mathbf{R}(\mathbf{z}) \mathbf{R}^{\dagger}(\mathbf{z}) + \mathbb{Z}(\mathbf{z}) \mathbb{Z}^{\dagger}(\mathbf{z}))]
$$

In a more compact form we have:

$$
\phi_k(\mathbf{R}(\mathbf{z}), \mathbb{Z}(\mathbf{z})) = \log E[\exp(\mathbf{x}(\mathbf{z})^2)]
$$

(2.1)

In the same way, the second generating function of the observations $\Phi_y$ taken at the point $\mathbf{w}$ of $\mathbb{C}^N$ is actually defined in $\mathbb{R}^{2N}$ by:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) \overset{\text{def}}{=} \log E[\exp(\mathbf{R}(\mathbf{w})^H \mathbf{R}(\mathbf{w}) + \mathbb{Z}(\mathbf{w})^H \mathbb{Z}(\mathbf{w}))]
$$

Thus we have:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) = \log E[\exp(\mathbf{w}^H \mathbf{H}^H \mathbf{w})]
$$

Replacing $y$ by its model yields:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) = \log E[\exp(\mathbf{w}^H \mathbf{H}^H \mathbf{H}_k \mathbf{w})]
$$

and from the sources mutual statistical independence hypothesis we can deduce:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) = \sum_k \log E[\exp(\mathbf{w}^H \mathbf{H}_k^H \mathbf{S}_k)]
$$

where $\mathbf{h}_k$ is the $k$th column of matrix $\mathbf{H}$. Then, (2.1) yields:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) = \sum_k \phi_k(\mathbf{R}(\mathbf{w}^T \mathbf{h}_k^H), \mathbb{Z}(\mathbf{w}^T \mathbf{h}_k^H))
$$

Finally, we define two real matrices $\mathbf{A}$ and $\mathbf{B}$ so that $\mathbf{H} = \mathbf{A} + j\mathbf{B}$. This leads to the new core equation that copes with the complex case:

$$
\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w})) = \sum_k \phi_k(\mathbf{A}_{nk} \mathbf{R}(\mathbf{w}_k) + \mathbf{B}_{nk} \mathbb{Z}(\mathbf{w}_k) + \mathbf{B}_{nk} \mathbb{Z}(\mathbf{w}_k) - \mathbf{B}_{nk} \mathbf{R}(\mathbf{w}_k))
$$

(2.2)

Note that defining $\phi_k, \Phi_y$ in $\mathbb{R}^{2N}$ and $\mathbb{R}^2$ respectively instead of $\mathbb{C}^N$ and $\mathbb{C}^2$ allows their differentiation. Hence, the next step is the differentiation of (2.2).

### 2.4 Differentiation of $\Phi_y(\mathbf{R}(\mathbf{w}), \mathbb{Z}(\mathbf{w}))$

We define $\mathbf{u} = \mathbf{R}(\mathbf{w}), \mathbf{v} = \mathbb{Z}(\mathbf{w})$ and $w = (\mathbf{u}, \mathbf{v})$. $w$ is an element of $\mathbb{R}^{2N}$ and (2.2) can be rewritten as:

$$
\Phi_y(\mathbf{w}) = \sum_k \phi_k(\mathbf{A}_{nk} \mathbf{u}_n + \mathbf{B}_{nk} \mathbf{v}_n, \sum_k \mathbf{A}_{nk} \mathbf{v}_n - \mathbf{B}_{nk} \mathbf{u}_n)
$$

(2.3)

We also introduce three functions $g_1, g_2$ and $g$ respectively defined by:

$$
g_1(\mathbf{w}) = \sum_n \mathbf{A}_{nk} \mathbf{u}_n + \mathbf{B}_{nk} \mathbf{v}_n \quad g_2(\mathbf{w}) = \sum_n \mathbf{A}_{nk} \mathbf{v}_n - \mathbf{B}_{nk} \mathbf{u}_n
$$

$$
g : \mathbb{R}^{2N} \rightarrow \mathbb{R}^2 \quad \mathbf{w} \mapsto g(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w}))
$$

The $\phi_k$ functions map $\mathbb{R}^2$ to $\mathbb{R}$ and we have:

$$
\Phi_y(\mathbf{w}) = \sum_k \phi_k(g(\mathbf{w}))
$$

Let us compute the partial derivatives of $\Phi_y(\mathbf{w})$ with respect to the real $(u_p, n = 1 \cdots N)$ and imaginary parts $(v_n, n = 1 \cdots N)$ of $\mathbf{w}$ coordinates. The differentiations of (2.3) with respect to $u_p, v_p, p = 1 \cdots N$ yield:

$$
\frac{\partial \Phi_y(\mathbf{w})}{\partial u_p} = \sum_{k=1}^K \frac{\partial \phi_k(g)}{\partial g_1} A_{pk} - \frac{\partial \phi_k(g)}{\partial g_2} B_{pk}
$$

(2.4)

$$
\frac{\partial \Phi_y(\mathbf{w})}{\partial v_p} = \sum_{k=1}^K \frac{\partial \phi_k(g)}{\partial g_1} B_{pk} + \frac{\partial \phi_k(g)}{\partial g_2} A_{pk}
$$

(2.5)

In order to have a sufficient diversity of equations we have to use higher differentiating orders. In the theoretical part of this study, we limit ourselves to the second order. The associated equations at higher orders can be obtained in a similar manner. Hence, we can differentiate (2.4) and (2.5) with respect to $u_q$ and $v_q, q = 1 \cdots N$. For instance,

$$
\frac{\partial^2 \Phi_y(\mathbf{w})}{\partial u_p \partial u_q} = \frac{\partial}{\partial u_q} \sum_{k=1}^K \frac{\partial \phi_k(g)}{\partial u_q} A_{pk} - \frac{\partial}{\partial u_q} \sum_{k=1}^K \frac{\partial \phi_k(g)}{\partial u_q} B_{pk}
$$

(2.6)
Substituting (2.4) and (2.5) in (2.6) yields:

\[
\begin{align*}
\frac{\partial^2 \Phi_p}{\partial u_p \partial u_q} &= \sum_{k=1}^{K} A_{pk} \left( \frac{\partial^2 \Phi_y}{\partial g_1 \partial g_1} A_{pk} - \frac{\partial^2 \Phi_y}{\partial g_1 \partial g_2} B_{pk} \right) \\
- \sum_{k=1}^{K} B_{pk} \left( \frac{\partial^2 \Phi_y}{\partial g_2 \partial g_1} A_{pk} - \frac{\partial^2 \Phi_y}{\partial g_2 \partial g_2} B_{pk} \right)
\end{align*}
\]

2.5 Tensor storage and decomposition

In practice, the partial derivatives of \( \Phi_t \) are computed at 5 points of \( \mathbb{R}^{2N} \) denoted \( w(t) \). The objective is again to increase the order of the tensor, aiming at achieving a better estimation quality. Let us define \( G_{ij}^k = \frac{\partial^2 \Phi_y}{\partial g_i \partial g_j} \) \( i = 1,2 \); \( j = 1,2 \) (one can note that \( G_{11}^k = G_{22}^k \)). This leads to the three distinct relations:

\[
\begin{align*}
\frac{\partial^2 \Phi_p(w(t))}{\partial u_p \partial u_q} &= \sum_{k=1}^{K} A_{pk} A_{qk} G_{11}^k - \sum_{k=1}^{K} A_{pk} B_{qk} G_{12}^k \\
&\quad + \sum_{k=1}^{K} B_{pk} A_{qk} G_{21}^k + \sum_{k=1}^{K} B_{pk} B_{qk} G_{22}^k \\
&= \sum_{k=1}^{K} G_{sk}^k (2.7)
\end{align*}
\]

\[
\begin{align*}
\frac{\partial^2 \Phi_p(w(t))}{\partial v_p \partial v_q} &= \sum_{k=1}^{K} B_{pk} A_{qk} G_{11}^k + \sum_{k=1}^{K} B_{pk} B_{qk} G_{12}^k \\
&\quad + \sum_{k=1}^{K} A_{pk} A_{qk} G_{21}^k + \sum_{k=1}^{K} A_{pk} B_{qk} G_{22}^k \\
&= \sum_{k=1}^{K} G_{sk}^k (2.8)
\end{align*}
\]

\[
\begin{align*}
\frac{\partial^2 \Phi_p(w(t))}{\partial u_p \partial v_q} &= \sum_{k=1}^{K} A_{pk} B_{qk} G_{11}^k + \sum_{k=1}^{K} A_{pk} A_{qk} G_{12}^k \\
&\quad - \sum_{k=1}^{K} B_{pk} A_{qk} G_{21}^k - \sum_{k=1}^{K} B_{pk} B_{qk} G_{22}^k \\
&= \sum_{k=1}^{K} G_{sk}^k (2.9)
\end{align*}
\]

Since all values of \( p \) and \( q \) are taken into consideration, equations (2.7-2.9) cover all the partial second order derivatives. In the real case, the second order derivatives of \( \Phi_t \) are stored in a third order tensor whose CanD gives a direct estimation of the mixing matrix. This situation is quite different in the complex case. Indeed, each of the three previous equations can be seen as a sum of four CanD of third-order tensors \( (p,q,s) \), involving the elements of the mixing matrix in different ways. It appears that the CanD of these tensors or of any combination of those is insufficient here. Therefore CanD based algorithms such as ALESCAF are not pertinent in this case. However it is still possible to use a tensor approach by jointly exploiting the three forms of derivatives in order to build a fourth-order tensor \( (N,N,S,3) \) with increased diversity, noted \( \mathcal{T} \). The last mode of \( \mathcal{T} \) contains the following elements:

\[
\begin{align*}
\mathcal{T}_{pqs1} &\text{ def } \frac{\partial^2 \Phi_p}{\partial u_p \partial u_q} \\
\mathcal{T}_{pqs2} &\text{ def } \frac{\partial^2 \Phi_p}{\partial v_p \partial v_q} \\
\mathcal{T}_{pqs3} &\text{ def } \frac{\partial^2 \Phi_p}{\partial u_p \partial v_q}
\end{align*}
\]

3. ALGORITHM

3.1 Building \( \mathcal{T} \)

We explain in this section how to build \( \mathcal{T} \) from the realizations of \( y \).

The entries of \( \mathcal{T} \) are computed one by one just like in the real case. We call \( \Gamma_3 \) the first generating function of \( y \) defined by:

\[
\Gamma_3(y^{(i)}) = \text{exp}([u^{(i)}]T \mathcal{R} \{y\} + v^{(i)}T \mathcal{S} \{y\}])
\]

Thus, the elements of \( \mathcal{T} \) (i.e. second order derivatives) are given by:

\[
\begin{align*}
\mathcal{T}_{pqs1} &= \frac{\partial \mathcal{T}_{pqs2}}{\partial \Phi_p} = \frac{\partial \mathcal{T}_{pqs1}}{\partial \Phi_q} \\
\mathcal{T}_{pqs2} &= \frac{\partial \mathcal{T}_{pqs2}}{\partial \Phi_p} = \frac{\partial \mathcal{T}_{pqs1}}{\partial \Phi_q} \\
\mathcal{T}_{pqs3} &= \frac{\partial \mathcal{T}_{pqs3}}{\partial \Phi_p} = \frac{\partial \mathcal{T}_{pqs1}}{\partial \Phi_q}
\end{align*}
\]

3.2 Estimation of the mixing matrix

The proposed algorithm is named LEMACAF-C, where \( O \) is the order of differentiation. Hence, LEMACAF-C-2 consists of iteratively fitting the tensor \( \mathcal{T} \) built from the estimated parameters and model equations (2.7-2.9) to \( \Phi \) using the Levenberg-Marquardt (LM) method. The LM method has been used to perform the CanD of multi-way arrays in [14,15] for example.

We consider the minimization of the following quadric cost function:

\[
\gamma_{\mathcal{T}}(p) = \frac{1}{2} \| e(p) \|_F^2 = \frac{1}{2} e^T(p)e(p)
\]

where \( e(p) = \text{vec} (\mathcal{T} \Phi - \mathcal{T} \Phi) \in \mathbb{C}^{2N^2} \times 1 \) is the residue and \( p \) is the parameter vector defined as:

\[
p = \frac{\text{vec}(\hat{A})}{\text{vec}(B)} = \left[ \begin{array}{c} \text{vec}(A) \\
\text{vec}(B) \\
\text{vec}(G^{11}) \\
\text{vec}(G^{12}) \\
\text{vec}(G^{21}) \\
\text{vec}(G^{22}) \end{array} \right] \in \mathbb{C}^{(2N^2+S+3) \times 1}
\]
Table 1: Some comparisons between LEMACAFC algorithms (denoted LEMC here) and 6-BIOME. Ns is the simulation number. NC means that the corresponding algorithm has never converged; × means that it has not been evaluated in this situation.

<table>
<thead>
<tr>
<th>Ns</th>
<th>Simulation parameters</th>
<th>Median values ($10^{-2}$) and number of acceptable values (%) of $f_H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4-QAM</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4-QAM</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4-QAM</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4-QAM</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>4-QAM</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4-QAM</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>4-QAM</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>8-PSK</td>
<td>4</td>
</tr>
</tbody>
</table>

and where \( \text{vec} \{ \cdot \} \) yields a column vector by stacking the columns of its matrix argument. The LM update is given as follows:

\[
p(i+1) = p(i) - [J^H(i)J(i) + \lambda(i)I]^{-1}g(i)
\]

where \( J(i) \) denotes the Jacobian matrix, \( g(i) \) the gradient vector computed at iteration \( i \) and \( \lambda(i) \) is a positive regularization parameter. \( p \) and \( \lambda \) are updated at every iteration. There are many ways to proceed with the LM updates. We retained the scheme described in [16]. After convergence, an estimate of the mixture is obtained by \( \hat{H} = \text{unvec} \{ \hat{p}_A + j\hat{p}_B \} \) (up to column permutation and scaling).

4. COMPUTER RESULTS

![Figure 1: Distribution (in %) of the relative estimation error ($f_H$) computed from simulation 4 results.](image)

The performance of the proposed approach is evaluated for blind identification of underdetermined complex mixtures of 4-QAM or 8-PSK sources. Our goal here is to highlight the performance of LEMACAFC through a limited number of key simulations.

Three versions of the algorithm (LEMACAFC-2, LEMACAFC-3, LEMACAFC-4) have been implemented and compared in various situations to the well known 6-BIOME (Blind Identification of Overcomplete Mixtures) algorithm [2], also referred to as “BIRTH” (Blind Identification of mixtures of sources using Redundancies in the data Hexacovariance matrix).

Algorithms were evaluated with respect to the estimation error, according to the following normalized measure:

\[
f_H(\hat{H}, H) = \frac{\text{vec}(\hat{H} - H)^H\text{vec}(\hat{H} - H)}{\text{vec}(H)^H\text{vec}(H)}
\]

For each situation a median value (Med.) of \( f_H \) and a number of ”acceptable results” (Na) is obtained from 50 Monte-Carlo runs. We chose to define Na as the percentage of Monte-Carlo results for which \( f_H < 10^{-2} \). These values are reported in Table 1 according to simulation parameters and algorithms. Figure 1 focuses on simulation 4 results, giving all LEMACAFC-3, LEMACAFC4 and 6-BIOME results in an histogram form. At each run, the source vectors and the mixing matrix were changed and the derivatives were computed at 10 different points (\( S = 10 \)) whose real and imaginary parts were randomly drawn in the range \([-1;1[^S \). Our iterative algorithms were all initialized with the same random entries and consistently stopped after 60 iterations. Simulation parameters are source modulation (Mod.), source number (\( K \)), sample number (\( M \)) and Signal to Noise Ratio (SNR). The number of sensors is fixed to 3 for each simulations. Three variations of LEMACAFC-2 were actually computed with an eye to evaluate the impact of an ”unlucky” initialization. The first one uses only one random initialization while for the second and third ones we compared five and ten different initializations respectively and we kept the initialization corresponding to the smallest value of \( f_H(p) \) after the 60 iterations. Only one random initialization was used for LEMACAFC-3 and LEMACAFC-4. Simulation 3 results show that increasing the number of random initialization appreciably improves our two performance criteria. Our simulations on 4-QAM sources can be rank in several categories. First of all, LEMACAFC-2 only converge in the most favourable situations (simulations 1,2,3). Actually, LEMACAFC-2 seems to be not suitable when the number of sources exceeds 4. Our simulations are ordered in the as-
cending order of LEMACAF-3 median values. Hence it clearly appears that for simulations 1 to 3 (i.e.: the easiest cases) LEMACAF-3 provides slightly better results than LEMACAF-4 as opposed to simulations 4 and 5 (middle cases). Finally LEMACAF-4 is sensibly better for simulations 6 and 7 (difficult cases), indicating that the latter is still interesting in some difficult situations, notably when the underdeterminacy level (i.e.: the ratio between source number and sensor number) is high.

Taking into account both criteria, LEMACAF-3 provides better or comparable results than 6-BIOME in most situations (simulations 1,2,3,5,7) while LEMACAF-4 is consistently better than the cumulant based approach at the exception of simulation 4, for which 6-BIOME provides 46% of acceptable values against 40% for LEMACAF-3 and LEMACAF-4. However the histogram plotted in figure 1 shows that when converging, LEMACAF provides a better estimation of the mixing matrix. For instance, 14% of LEMACAF-4 error values are smaller than 0.0033 against 8% for LEMACAF-3 and only 4% for 6-BIOME. In this sense, figure 1 is typical because a similar observation could have been done from every simulation histogram. Furthermore the number of LEMACAF-3 and LEMACAF-4 acceptable values could be increase by trying several random initialization entries.

Finally simulation 8 shows that in this "easy" case all LEMACAF algorithms provide satisfactory results with 8-PSK sources as opposed to 6-BIOME.

5. CONCLUSION

We have addressed the problem of blind identification of underdetermined complex mixtures of complex sources using the second generating function of the observations. We detailed the theoretical background and proposed an algorithm relying on an original tensor decomposition. Finally, three versions of this algorithm, based on several differentiation order, have been evaluated on simulated complex mixtures of telecommunications complex sources.

It has been shown that second order version provides some satisfying results in the least difficult cases, especially if several initialization entries are compared. In these conditions, it can be an option if one is looking for a fast algorithm. On the contrary, the fourth order version appears as a possible solution for the most complicated cases. In this connection, a deeper investigation should clarify the respective influences of SNR, underdeterminacy level and sample number. In other cases we recommend the use of LEMACAF-3 which is enough to overpass a classical 6 order cumulant based approach in most situations while being less time consuming than LEMACAF-4.

Moreover, the LEMACAF algorithm also worked fine in the case of 8-PSK sources.

REFERENCES


