

# Blind Identification of Underdetermined Mixtures Based on the Characteristic Function: The Complex Case

Xavier Luciani\*, *Member, IEEE*, André L. F. de Almeida, *Member, IEEE*, and Pierre Comon, *Fellow, IEEE*

**Abstract**—Blind identification of underdetermined mixtures can be addressed efficiently by using the second ChAracteristic Function (CAF) of the observations. Our contribution is twofold. First, we propose the use of a Levenberg-Marquardt algorithm, herein called LEMACAF, as an alternative to an Alternating Least Squares algorithm known as ALESCAF, which has been used recently in the case of real mixtures of real sources. Second, we extend the CAF approach to the case of complex sources for which the previous algorithms are not suitable. We show that the complex case involves an appropriate tensor stowage, which is linked to a particular tensor decomposition. An extension of the LEMACAF algorithm, called LEMACAF<sub>C</sub> is then proposed to blindly estimate the mixing matrix by exploiting this tensor decomposition. In our simulation results, we first provide performance comparisons between third- and fourth- order versions of ALESCAF and LEMACAF in various situations involving BPSK sources. Then, a performance study of LEMACAF<sub>C</sub> is carried out considering 4-QAM sources. These results show that the proposed algorithm provides satisfying estimations especially in the case of a large underdeterminacy level.

**Index Terms**—Blind identification, blind source separation, characteristic function, complex sources, underdetermined mixtures, tensor decompositions

## I. INTRODUCTION

**B**LIND Identification of linear mixtures (BI) has now become a major area of signal processing. For instance, since the theory of Independent Component Analysis (ICA) [1], this subject has been at the center of many theoretical works while related methods and algorithms have been successfully used in applicative fields, notably in telecommunications [2], acoustic [3] or biomedical engineering [4], [5] among others; see [6], [7] for surveys.

In the meantime, tensor analysis has gained attention in numerous application areas involving data analysis such as Psychometrics [8], Arithmetic Complexity [9] and Chemometrics [10], [11]. In particular, the Canonical Decomposition (CanD) [8] also known as PARAllel FACtor analysis (PARAFAC) [12] has met with success. One of the reasons is that CanD

can often favorably replace Principal Component Analysis (PCA), when available data measurements can be arranged in a meaningful tensor form [13]. Indeed, the CanD comes with a nice uniqueness property [14]–[18] and some simple numerical algorithms [10], [19], [20].

Due to multiple connections between the two areas, these advantages have been rapidly exploited for BI purposes [21]–[24]. In addition, tensor-based algorithms allow to solve the problem of underdetermined mixtures (*i.e.* when the number of sources is greater than the number of sensors), which arises in many practical situations, especially in telecommunications, and in which we are presently interested. A first class of algorithms exploits the trilinear nature of the observations, and the CanD of the data tensor provides a direct source estimation. For instance, this deterministic approach is widely used in fluorescence spectroscopy [10], [11]. When the observation diversity is not sufficient, one can resort to a second class of algorithms, using the multilinearity properties of High-Order Statistics (HOS) [20]. A large majority of these algorithms involves a tensor containing the cumulants of the observations, the decomposition of which leads to the blind identification of the mixing matrix [23], [25]. This is notably the case of FOObI [26], FOObI2 [22] and 6-BIOME [27] algorithms, which use 4th and 6th order cumulant tensors, respectively. Nevertheless, a different class of BI methods not exploiting cumulants but the second ChAracteristic Function (CAF) of the observations, has been proposed in [28]–[30]. We are particularly interested in the approach originally proposed in [29], leading to efficient algorithms such as ALESCAF [30]. In that work, the authors showed that partial derivatives of the second characteristic function can be stored in a symmetric tensor, the CanD of which provides a direct estimation of the mixing matrix up to trivial scaling and permutation indeterminacies. In [30], the ALESCAF algorithm is applied to a data tensor constructed from third-order derivatives of the characteristic function. It is worth mentioning that the ALESCAF algorithm has only been applied to BI problems involving real sources (*e.g.* BPSK and 4-PAM). The present study notably generalizes the CAF approach to the case of complex mixtures of complex sources, which often occurs in digital communications and for which the ALESCAF algorithm needs non trivial extensions.

The paper is organized as follows. In Section II, the BI problem is formulated and the CAF approach is briefly presented by first considering the case of real sources. A new LEMACAF algorithm that copes with the real case is

Xavier Luciani is with the french institute INSERM U642 and with the University of Rennes 1, LTSI, Campus de Beaulieu, bât. 22, 35042 Cedex, Rennes, FRANCE. E-mail: lucianix@gmail.com. André L. F. de Almeida is with Wireless Telecom Research Group, Department of Teleinformatics Engineering, Federal University of Ceará, CP 6005, Campus do Pici, 60455-760, Fortaleza, Ceará, Brazil. E-mail: andre@gtel.ufc.br. Pierre Comon is with Lab. Informatique Signaux et Systèmes de Sophia-Antipolis (I3S), UMR6070 CNRS-UNS, 2000 route des Lucioles, BP.121, F06903 Sophia Antipolis cedex, FRANCE, and with INRIA, Galaad, 2004 route des Lucioles, BP.93, F06902 Sophia Antipolis cedex, France. E-mail: pcomon@unice.fr

also introduced in this section. In Section III, we transpose the CAF approach to the case of complex sources. A new core equation is obtained and an appropriate decomposition of the derivative tensor is detailed. In order to implement this more general approach, we propose a suitable algorithm called LEMACAF $\mathbb{C}$  in Section IV. Computer simulation results considering both the real and complex cases are reported respectively in Sections V and VI. The paper is concluded in Section VII. Matlab codes including notably ALESCAF, LEMACAF and LEMACAF $\mathbb{C}$  algorithms can be found at <http://www.i3s.unice.fr/~pcomon/TensorPackage.html>.

*Notations:* In the following, 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  $i^{\text{th}}$  coordinate of vector  $\mathbf{a}$  and  $\mathbf{a}_i$  is the  $i^{\text{th}}$  column 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.  $\mathbb{E}[\cdot]$  denotes the expected value of a random variable.

## II. PROBLEM FORMULATION AND CAF APPROACH IN THE REAL CASE

We consider here the classical linear model of a noisy mixture of  $K$  stationary sources received by an array of  $N$  sensors. The mixture is instantaneous and under-determined ( $N < K$ ) and defined by a mixing matrix  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_K] \in \mathbb{R}^{N \times K}$ . Define also  $\mathbf{z}(m) = [z_1(m), \dots, z_N(m)]^T \in \mathbb{R}^N$ ,  $\mathbf{s}(m) = [s_1(m), \dots, s_K(m)]^T \in \mathbb{R}^K$  and  $\mathbf{n}(m) \in \mathbb{R}^N$  as the  $m^{\text{th}}$  realizations of the observation, source and noise vectors, respectively,  $m = 1, \dots, M$ . According to this linear model we have:

$$\mathbf{z}(m) = \mathbf{H}\mathbf{s}(m) + \mathbf{n}(m).$$

Algorithms from the CAF family use the partial derivatives of the observations characteristic function to identify the mixing matrix  $\mathbf{H}$  under the following assumptions:

- H1.** The mixing matrix  $\mathbf{H}$  does not contain pairwise collinear columns.
- H2.** The sources  $s_1, \dots, s_K$  are non-Gaussian and mutually statistically independent.
- H3.** The number of sources  $K$  is known.

It has been shown in former studies [31] [32] that  $\mathbf{H}$  is theoretically identifiable under these assumptions.

Here, we briefly recall the main steps of the CAF approach originally proposed in [29]. Let us denote  $\Phi_z$  and  $\varphi_k$  the second generating functions<sup>1</sup> of the observations and source  $k$  respectively:

$$\begin{aligned} \varphi_k(x) &\stackrel{\text{def}}{=} \log \mathbb{E}[\exp(x s_k)], \quad x \in \mathbb{R}, \\ \Phi_z(\mathbf{u}^{(s)}) &\stackrel{\text{def}}{=} \log \mathbb{E}[\exp(\mathbf{u}^{(s)T} \mathbf{z})], \quad \mathbf{u}^{(s)} \in \mathbb{R}^N. \end{aligned}$$

Replacing  $\mathbf{z}$  by its model and neglecting the noise contribution leads to the decomposition of the observation generating

function into the sum of the sources individual generating functions:

$$\Phi_z(\mathbf{u}^{(s)}) = \log \mathbb{E} \left[ \exp \left( \sum_{n,k} u_n^{(s)} H_{nk} s_k \right) \right].$$

Using the source independence property, we get:

$$\Phi_z(\mathbf{u}^{(s)}) = \sum_k \varphi_k \left( \sum_n H_{nk} u_n^{(s)} \right). \quad (1)$$

Equation (1) is the core equation of the CAF approach in the real field. Differentiating (1)  $P$  times with respect to  $P$  components of  $\mathbb{R}^N$ , denoted  $u_{n_1}, u_{n_2}, \dots, u_{n_P}$ , and defining  $G_{sk} \stackrel{\text{def}}{=} \frac{\partial^P \varphi_k(\sum_n H_{nk} u_n^{(s)})}{\partial u_{n_1} \partial u_{n_2} \dots \partial u_{n_P}}$  we obtain:

$$\frac{\partial^P \Phi_z(\mathbf{u}^{(s)})}{\partial u_{n_1} \partial u_{n_2} \dots \partial u_{n_P}} = \sum_{k=1}^K H_{n_1 k} H_{n_2 k} \dots H_{n_P k} G_{sk}, \quad (2)$$

with  $n_p = 1, \dots, N$  and  $p = 1, \dots, P$ . These derivatives could be stored in a  $P$ -th order tensor but in practice, the partial derivatives of  $\Phi_z$  are computed in  $S$  points ( $\mathbf{u}^{(1)} \dots \mathbf{u}^{(S)}$ ) of  $\mathbb{R}^N$ . The objective is to increase the order of the tensor, aiming at achieving a better estimation quality. Hence, we now have a  $(P+1)$ -th order data tensor, the last dimension describing the  $S$  differentiation points. The key issue of the CAF approach is that (2) is nothing else but the rank- $K$  (truncated) CanD of the data tensor, which allows the identification of matrix  $\mathbf{H}$ . Indeed, when the number of sources is smaller than the generic rank of the tensor, this decomposition admits an essentially unique solution for  $\mathbf{H}$  and  $\mathbf{G}$ , (*i.e.* up to scaling and permutation of their columns), where  $\mathbf{G}$  is the  $S \times K$  matrix with entries  $G_{sk}$ .

The general structure of CAF algorithms can be summarized as follows:

- 1) Choose  $S$  points of  $\mathbb{R}^N$ ;
- 2) Compute for each point order  $P$  partial derivatives of  $\Phi_z$  and store the results in a tensor  $\mathcal{T}$ ;
- 3) Estimate  $\mathbf{H}$  from the rank- $K$  decomposition of  $\mathcal{T}$ .

Note that the differentiation order  $P$  is an input parameter of the algorithm. The higher the differentiation order, the higher the tensor order, and hence its generic rank for these dimensions. Consequently, increasing the differentiation order should allow to identify mixtures involving a larger number of sources without increasing the number of sensors. The price to pay is, of course, an increase in the algorithm complexity and probably a loss in robustness and accuracy.

The ALESCAF algorithm resorts to a classical Alternating Least Squares (ALS) procedure in order to perform the CanD. Algorithm refinements can be added so as to improve the convergence speed and avoid local minima, such as the Enhanced Line Search (ELS) procedure [33], [20]. Other CanD algorithms can be used [19], [20]. In order to identify the most suitable algorithm in the CAF context, we have compared the ALS-ELS approach with the gradient-ELS descent and the Levenberg-Marquardt (LM) optimization methods for various tensors following model (2) (results not shown). Among these, the LM method has shown a good compromise in terms

<sup>1</sup>In order to simplify notations and calculations, without any theoretical impact, we prefer using the generating function instead of the characteristic function.

of convergence speed and estimation accuracy. In addition, the successful tensor-based applications of the LM method in different applications [19], [34], [35] has motivated us to introduce a LM-based algorithm called LEMACAF, which achieves the decomposition of higher-order tensors constructed from the derivatives of the characteristic function as follows: Define  $\widehat{\mathbf{H}}^{(i)}$  as the estimated matrix corresponding to the  $i^{\text{th}}$  mode of the CanD (2),  $i = 1, \dots, P$ , and  $\widehat{\mathbf{G}}$  as the estimated matrix corresponding to the  $(P+1)$ -th mode. Let  $\widehat{\mathcal{T}}(\widehat{\mathbf{H}}^{(1)}, \dots, \widehat{\mathbf{H}}^{(P)}, \widehat{\mathbf{G}})$  be the tensor built from the estimated matrices. Note that ideally we should have  $\widehat{\mathbf{H}}^{(1)} = \widehat{\mathbf{H}}^{(2)} = \dots = \widehat{\mathbf{H}}^{(P)} = \mathbf{H}$  and  $\widehat{\mathcal{T}} = \mathcal{T}$ . We consider the minimization of the following quadratic cost function:

$$f_{\Phi}(\mathbf{p}) = \frac{1}{2} \|\mathbf{e}_{\Phi}(\mathbf{p})\|_F^2 = \frac{1}{2} \mathbf{e}_{\Phi}^T(\mathbf{p}) \mathbf{e}_{\Phi}(\mathbf{p}), \quad (3)$$

where  $\mathbf{e}_{\Phi}(\mathbf{p}) = \text{vec}(\mathcal{T} - \widehat{\mathcal{T}}(\widehat{\mathbf{H}}^{(1)}, \dots, \widehat{\mathbf{H}}^{(P)}, \widehat{\mathbf{G}}))$  is the residue and  $\mathbf{p}$  is the parameter vector defined as:

$$\mathbf{p} = \left[ \begin{array}{cccc} (\text{vec}(\widehat{\mathbf{H}}^{(1)\text{T}}))^T & \dots & (\text{vec}(\widehat{\mathbf{H}}^{(P)\text{T}}))^T & (\text{vec}(\widehat{\mathbf{G}}^T))^T \end{array} \right]^T, \quad (4)$$

where  $\text{vec}(\cdot)$  maps a matrix or a tensor to a column vector by stacking its columns one below the other. The LM update at iteration  $k+1$  is given by:

$$\mathbf{p}(k+1) = \mathbf{p}(k) - [\mathbf{J}^T(k) \mathbf{J}(k) + \lambda(k) \mathbf{I}]^{-1} \mathbf{g}(k), \quad (5)$$

where  $\mathbf{J}$  denotes the Jacobian matrix given by:  $J_{ij}(k) = \frac{\partial e_{\Phi i}(\mathbf{p})}{\partial p_j}$ ,  $\mathbf{g}$  is the gradient vector given by  $\mathbf{g}(k) = \mathbf{J}(k)^T \mathbf{e}_{\Phi}(\mathbf{p})$ , or equivalently:  $g_j(k) = \frac{\partial f_{\Phi}(\mathbf{p})}{\partial p_j}$  and  $\lambda(k)$  is a positive regularization parameter. At every iteration  $k$ ,  $\mathbf{g}$ ,  $\mathbf{J}$ ,  $\mathbf{p}$  and  $\lambda$  are updated. There are many ways to proceed, and we retained the scheme described in [36]:

- 1) Compute  $\Delta_p(k) = -[\mathbf{J}^T(k) \mathbf{J}(k) + \lambda(k) \mathbf{I}]^{-1} \mathbf{g}(k)$
- 2) Compute  $\mathbf{p}(k+1)$  and deduce  $f_{\Phi}(\mathbf{p}(k+1))$
- 3) Compute  $\Upsilon = |f_{\Phi}(\mathbf{p}(k+1)) - f_{\Phi}(\mathbf{p}(k))| \cdot |\hat{f}_{\Phi}(\mathbf{p}(k+1)) - f_{\Phi}(\mathbf{p}(k))|^{-1}$  where  $\hat{f}_{\Phi} = f_{\Phi}(\mathbf{p}) + \mathbf{d}_p^T \mathbf{J}^T \mathbf{e}_{\Phi} + \frac{1}{2} \mathbf{d}_p^T \mathbf{J}^T \mathbf{J} \mathbf{d}_p$  is the second order approximation of  $f_{\Phi}$ .
- 4) if  $\Upsilon \geq 0$  then  $\mathbf{p}(k+1)$  is accepted,  $\lambda(k+1) = \lambda(k) * \max(\frac{1}{3}, 1 - (2\Upsilon - 1)^3)$  and  $\nu = 2$ . Otherwise  $\mathbf{p}(k+1)$  is rejected,  $\lambda(k+1) = \nu \lambda(k)$  and  $\nu = 2 * \nu$ .

Compact forms of the gradient vector and Jacobian matrix for a third order tensor can be found in [34] and [20]. Those can be easily generalized for higher order tensors. After convergence, an estimate  $\widehat{\mathbf{H}}$  of the mixture is obtained from the average of  $\widehat{\mathbf{H}}^{(1)}, \dots, \widehat{\mathbf{H}}^{(P)}$  after a column-wise normalization.

It is worth mentioning that, although ELS refinements and symmetric constraints are applicable to improve the convergence speed of the LEMACAF algorithm, our preliminary numerical simulation study has shown no significant improvement. Therefore, these refinements are not considered here. Order 3 and 4 versions of the LEMACAF algorithm will be considered later in Section V.

### III. EXTENSION TO THE COMPLEX FIELD

In this section, we generalize the BI problem based on the characteristic function to the complex field, i.e. to the case of complex mixtures of complex sources. Although the

theoretical aspects are similar to the real case, the characteristic function core equation (1) cannot be used directly in the complex field therefore the CanD of the derivatives given in (2) is no more valid in the complex sources case.

The generalization of the CAF approach to the complex case involves the following steps: i) choosing an appropriate core equation, ii) deduce the associated tensor decomposition by differentiating this core equation and, finally, iii) formulating an efficient algorithm to estimate the mixing matrix from the structure of the obtained tensor decomposition. In this section we address the two first steps.

#### A. The new core equation

Observation  $\underline{\mathbf{z}}$  and source  $\underline{\mathbf{s}}$  vectors belong now to the complex field as well as the mixing matrix  $\underline{\mathbf{H}}$ .

The second generating function of the observations,  $\tilde{\Phi}_z$ , can still be decomposed into a sum of marginal second generating functions of sources,  $\tilde{\varphi}_k$ ,  $k = 1 \dots K$ . In order to see this, start from the definitions of  $\tilde{\Phi}_z$  and  $\tilde{\varphi}_k$  in the complex field. Generating functions of a complex variable are actually defined by assimilating  $\mathbb{C}$  to  $\mathbb{R}^2$ . Thus the second generating function of the  $k^{\text{th}}$  source  $\tilde{\varphi}_k$  taken at the point  $\underline{\mathbf{l}}$  of  $\mathbb{C}$  is defined as a function of the real and imaginary parts of  $\underline{\mathbf{l}}$ :

$$\tilde{\varphi}_k(\Re\{\underline{\mathbf{l}}\}, \Im\{\underline{\mathbf{l}}\}) \stackrel{\text{def}}{=} \log E [\exp(\Re\{\underline{\mathbf{s}}_k\} \Re\{\underline{\mathbf{l}}\} + \Im\{\underline{\mathbf{s}}_k\} \Im\{\underline{\mathbf{l}}\})].$$

In a more compact form we have:

$$\tilde{\varphi}_k(\Re\{\underline{\mathbf{l}}\}, \Im\{\underline{\mathbf{l}}\}) = \log E [\exp(\Re\{\underline{\mathbf{l}}^* \underline{\mathbf{s}}_k\})]. \quad (6)$$

This bijection also applies to  $\tilde{\Phi}_z$ . Hence,  $\tilde{\Phi}_z$  taken at the point  $\underline{\mathbf{w}}$  of  $\mathbb{C}^N$  is actually defined in  $\mathbb{R}^{2N}$  by

$$\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\}) \stackrel{\text{def}}{=} \log E [\exp(\mathbf{x}^T \Re\{\underline{\mathbf{w}}\} + \mathbf{y}^T \Im\{\underline{\mathbf{w}}\})],$$

where  $\mathbf{x} = \Re\{\underline{\mathbf{z}}\}$  and  $\mathbf{y} = \Im\{\underline{\mathbf{z}}\}$  and thus we have

$$\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\}) = \log E [\exp(\Re\{\underline{\mathbf{w}}^H \underline{\mathbf{z}}\})].$$

Now, replacing  $\underline{\mathbf{z}}$  by its model and neglecting the noise contribution yields:

$$\begin{aligned} \tilde{\Phi}_z(\underline{\mathbf{w}}) &= \log E [\exp(\Re\{\underline{\mathbf{w}}^H \underline{\mathbf{H}} \underline{\mathbf{s}}\})], \\ &= \log E \left[ \exp(\Re\{\underline{\mathbf{w}}^H \sum_k \underline{\mathbf{h}}_k \underline{\mathbf{s}}_k\}) \right], \\ &= \log E \left[ \prod_k \exp(\Re\{\underline{\mathbf{w}}^H \underline{\mathbf{h}}_k \underline{\mathbf{s}}_k\}) \right], \end{aligned}$$

where  $\underline{\mathbf{h}}_k$  is the  $k^{\text{th}}$  column of  $\underline{\mathbf{H}}$ . Then, using the sources mutual statistical independence hypothesis we can deduce:

$$\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\}) = \sum_k \log E [\exp(\Re\{\underline{\mathbf{w}}^H \underline{\mathbf{h}}_k \underline{\mathbf{s}}_k\})]$$

and (6) yields:

$$\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\}) = \sum_k \tilde{\varphi}_k(\Re\{\underline{\mathbf{w}}^T \underline{\mathbf{h}}_k^*\}, \Im\{\underline{\mathbf{w}}^T \underline{\mathbf{h}}_k^*\}).$$

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

$$\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\}) = \sum_k \tilde{\varphi}_k \left( \sum_n A_{nk} \Re\{\underline{w}_n\} + B_{nk} \Im\{\underline{w}_n\}, \sum_n A_{nk} \Im\{\underline{w}_n\} - B_{nk} \Re\{\underline{w}_n\} \right). \quad (7)$$

Note that defining  $\tilde{\Phi}_z$  and  $\tilde{\varphi}_k$  in  $\mathbb{R}^{2N}$  and  $\mathbb{R}^2$  respectively instead of  $\mathbb{C}^N$  and  $\mathbb{C}$  allows their differentiation. Hence, the next step is the differentiation of (7).

### B. Differentiation of $\tilde{\Phi}_z(\Re\{\underline{\mathbf{w}}\}, \Im\{\underline{\mathbf{w}}\})$

Let us define  $\mathbf{u} = \Re\{\underline{\mathbf{w}}\}$ ,  $\mathbf{v} = \Im\{\underline{\mathbf{w}}\}$  and  $\mathbf{w} = (\mathbf{u}, \mathbf{v})$ , so that  $\mathbf{w}$  belongs to  $\mathbb{R}^{2N}$ . From these definitions, (7) can be rewritten as:

$$\tilde{\Phi}_z(\mathbf{w}) = \sum_k \tilde{\varphi}_k \left( \sum_n A_{nk} u_n + B_{nk} v_n, \sum_n A_{nk} v_n - B_{nk} u_n \right), \quad (8)$$

where

$$\begin{aligned} \tilde{\Phi}_z : \mathbb{R}^{2N} &\longrightarrow \mathbb{R} \\ \mathbf{w} &\longmapsto \tilde{\Phi}_z(\mathbf{w}). \end{aligned}$$

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

$$\begin{aligned} g_1(\mathbf{w}) &= \sum_n A_{nk} u_n + B_{nk} v_n, \\ g_2(\mathbf{w}) &= \sum_n A_{nk} v_n - B_{nk} u_n, \end{aligned}$$

and

$$\begin{aligned} g : \mathbb{R}^{2N} &\longrightarrow \mathbb{R}^2 \\ \mathbf{w} &\longmapsto g(\mathbf{w}) = (g_1(\mathbf{w}), g_2(\mathbf{w})). \end{aligned}$$

Functions  $\tilde{\varphi}_k$  map  $\mathbb{R}^2$  to  $\mathbb{R}$ :

$$\begin{aligned} \tilde{\varphi}_k : \mathbb{R}^2 &\longrightarrow \mathbb{R} \\ g &\longmapsto \tilde{\varphi}_k(g). \end{aligned}$$

This yields a compact representation of (8) as follows:

$$\tilde{\Phi}_z(\mathbf{w}) = \sum_k \tilde{\varphi}_k(g(\mathbf{w})).$$

Now, we can compute the partial derivatives of  $\tilde{\Phi}_z(\mathbf{w})$  with respect to the real ( $u_n, n = 1 \dots N$ ) and imaginary ( $v_n, n = 1 \dots N$ ) parts of  $\underline{\mathbf{w}}$ . Similarly to the real case, in order to have a sufficient diversity of equations, we have to use higher differentiating orders. The objective is to increase the order of the tensor, with the goal of achieving a better estimation quality. In the theoretical part of this study, we limit ourselves to second and third orders, being understood that equations associated with higher differentiation orders can be obtained in a similar manner.

The number of equations can also be increased for a fixed differentiation order, by computing partial derivatives of  $\tilde{\Phi}_z$  in  $S$  different points of  $\mathbb{R}^{2N}$ , denoted here as  $\mathbf{w}^{(s)} = (u^{(s)}, v^{(s)})$ ,  $s = 1 \dots S$ .

1) *Order 2 derivatives:* At order 2, we obtain:

$$\begin{aligned} \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial u_q} &= \sum_{k=1}^K A_{pk} A_{qk} G_{sk}^{11} - \sum_{k=1}^K A_{pk} B_{qk} G_{sk}^{12} - \\ &\quad \sum_{k=1}^K B_{pk} A_{qk} G_{sk}^{12} + \sum_{k=1}^K B_{pk} B_{qk} G_{sk}^{22}. \quad (9) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p \partial v_q} &= \sum_{k=1}^K B_{pk} B_{qk} G_{sk}^{11} + \sum_{k=1}^K B_{pk} A_{qk} G_{sk}^{12} + \\ &\quad \sum_{k=1}^K A_{pk} B_{qk} G_{sk}^{12} + \sum_{k=1}^K A_{pk} A_{qk} G_{sk}^{22}, \quad (10) \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial v_q} &= \sum_{k=1}^K A_{pk} B_{qk} G_{sk}^{11} + \sum_{k=1}^K A_{pk} A_{qk} G_{sk}^{12} - \\ &\quad \sum_{k=1}^K B_{pk} B_{qk} G_{sk}^{12} - \sum_{k=1}^K B_{pk} A_{qk} G_{sk}^{22}. \quad (11) \end{aligned}$$

where

$$G_{sk}^{ij} = \frac{\partial^2 \tilde{\varphi}_k(g(\mathbf{w}^{(s)}))}{\partial g_i(\mathbf{w}^{(s)}) \partial g_j(\mathbf{w}^{(s)})}; \quad i = 1, 2; \quad j = 1, 2.$$

Thereby, each of the three second-order derivatives (9)-(11) are given by a sum of four different third-order CanDs involving the elements of the mixing matrix in different ways. Note that, since all values of  $p$  and  $q$  are taken into consideration, (9)-(11) cover all the partial second order derivatives. We show in Appendix A how to derive (9)-(11) from (8).

2) *Order 3 derivatives:* By differentiating both (9) and (10) with respect to  $u_r^{(s)}$  and  $v_r^{(s)}$ ,  $r = 1 \dots N$ , we can obtain the four different order 3 equations. Let us define

$$G_{sk}^{hij} = \frac{\partial^3 \tilde{\varphi}_k(g(\mathbf{w}^{(s)}))}{\partial g_h(\mathbf{w}^{(s)}) \partial g_i(\mathbf{w}^{(s)}) \partial g_j(\mathbf{w}^{(s)})}.$$

Using the fact that  $G_{sk}^{211} = G_{sk}^{121} = G_{sk}^{112}$  and  $G_{sk}^{221} = G_{sk}^{122} = G_{sk}^{212}$ , we get:

$$\begin{aligned} \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial u_q \partial u_r} &= \sum_{k=1}^K A_{pk} A_{qk} A_{rk} G_{sk}^{111} - \sum_{k=1}^K A_{pk} A_{qk} B_{rk} G_{sk}^{211} - \\ &\quad \sum_{k=1}^K A_{pk} B_{qk} A_{rk} G_{sk}^{211} - \sum_{k=1}^K B_{pk} A_{qk} A_{rk} G_{sk}^{211} + \\ &\quad \sum_{k=1}^K A_{pk} B_{qk} B_{rk} G_{sk}^{221} + \sum_{k=1}^K B_{pk} A_{qk} B_{rk} G_{sk}^{221} + \\ &\quad \sum_{k=1}^K B_{pk} B_{qk} A_{rk} G_{sk}^{221} - \sum_{k=1}^K B_{pk} B_{qk} B_{rk} G_{sk}^{222}, \quad (12) \end{aligned}$$

$$\begin{aligned} \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial u_q \partial v_r} = & \sum_{k=1}^K A_{pk} A_{qk} B_{rk} G_{sk}^{111} + \sum_{k=1}^K A_{pk} A_{qk} A_{rk} G_{sk}^{211} - \\ & \sum_{k=1}^K A_{pk} B_{qk} B_{rk} G_{sk}^{211} - \sum_{k=1}^K B_{pk} A_{qk} B_{rk} G_{sk}^{211} - \\ & \sum_{k=1}^K A_{pk} B_{qk} A_{rk} G_{sk}^{221} - \sum_{k=1}^K B_{pk} A_{qk} A_{rk} G_{sk}^{221} + \\ & \sum_{k=1}^K B_{pk} B_{qk} B_{rk} G_{sk}^{221} + \sum_{k=1}^K B_{pk} B_{qk} A_{rk} G_{sk}^{222}, \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p \partial v_q \partial v_r} = & \sum_{k=1}^K B_{pk} B_{qk} B_{rk} G_{sk}^{111} + \sum_{k=1}^K B_{pk} B_{qk} A_{rk} G_{sk}^{211} + \\ & \sum_{k=1}^K B_{pk} A_{qk} B_{rk} G_{sk}^{211} + \sum_{k=1}^K A_{pk} B_{qk} B_{rk} G_{sk}^{211} + \\ & \sum_{k=1}^K B_{pk} A_{qk} A_{rk} G_{sk}^{221} + \sum_{k=1}^K A_{pk} B_{qk} A_{rk} G_{sk}^{221} + \\ & \sum_{k=1}^K A_{pk} A_{qk} B_{rk} G_{sk}^{221} + \sum_{k=1}^K A_{pk} A_{qk} A_{rk} G_{sk}^{222}, \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p \partial v_q \partial u_r} = & \sum_{k=1}^K B_{pk} B_{qk} A_{rk} G_{sk}^{111} - \sum_{k=1}^K B_{pk} B_{qk} B_{rk} G_{sk}^{211} + \\ & \sum_{k=1}^K B_{pk} A_{qk} A_{rk} G_{sk}^{211} + \sum_{k=1}^K A_{pk} B_{qk} A_{rk} G_{sk}^{211} - \\ & \sum_{k=1}^K B_{pk} A_{qk} B_{rk} G_{sk}^{221} - \sum_{k=1}^K A_{pk} B_{qk} B_{rk} G_{sk}^{221} + \\ & \sum_{k=1}^K A_{pk} A_{qk} A_{rk} G_{sk}^{221} - \sum_{k=1}^K A_{pk} A_{qk} B_{rk} G_{sk}^{222}. \end{aligned} \quad (15)$$

Thus, in the case of order 3 derivatives, each equation is now given by a sum of eight fourth-order CanDs involving the elements of the mixing matrix in different ways.

### C. Tensor stowage and decomposition

As we have seen in Section II, in the real case the second order derivatives of  $\tilde{\Phi}_z$  can be stored in a third-order tensor, the CanD of which gives a direct estimation of the mixing matrix. The situation is quite different in the complex case but we still use a tensor approach to jointly exploit the different forms of derivatives. Let us first consider the case of second order derivatives. From (9)-(11), we propose to build a fourth-order tensor  $\mathcal{T}^{\tilde{\Phi}^2}$  of dimensions  $(N, N, S, 3)$  containing all the three derivative equations by concatenating all the associated decompositions, as follows:

$$\mathcal{T}_{pqrs1}^{\tilde{\Phi}^2} = \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p^{(s)} \partial u_q^{(s)}}; \quad \mathcal{T}_{pqrs2}^{\tilde{\Phi}^2} = \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p^{(s)} \partial v_q^{(s)}}; \quad (16)$$

$$\mathcal{T}_{pqrs3}^{\tilde{\Phi}^2} = \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p^{(s)} \partial v_q^{(s)}}. \quad (17)$$

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.

By applying the same reasoning to third order equations (12)-(15), we can build a fifth-order tensor  $\mathcal{T}^{\tilde{\Phi}^3}$  of dimensions  $(N, N, N, S, 3)$  as:

$$\mathcal{T}_{pqr s1}^{\tilde{\Phi}^3} = \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p^{(s)} \partial u_q^{(s)} \partial u_r^{(s)}}; \quad \mathcal{T}_{pqr s2}^{\tilde{\Phi}^3} = \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p^{(s)} \partial u_q^{(s)} \partial v_r^{(s)}}; \quad (18)$$

$$\mathcal{T}_{pqr s3}^{\tilde{\Phi}^3} = \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p^{(s)} \partial v_q^{(s)} \partial v_r^{(s)}}; \quad \mathcal{T}_{pqr s4}^{\tilde{\Phi}^3} = \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p^{(s)} \partial v_q^{(s)} \partial u_r^{(s)}}. \quad (19)$$

Our goal is to devise an algorithm capable of jointly estimating the real and imaginary parts  $\mathbf{A}$  and  $\mathbf{B}$  of the mixing matrix from  $\mathcal{T}^{\tilde{\Phi}^2}$  or  $\mathcal{T}^{\tilde{\Phi}^3}$ . This issue is addressed in the next section.

## IV. ALGORITHM FOR THE COMPLEX CASE

### A. Building the derivative tensor

First of all, we have to build  $\mathcal{T}^{\tilde{\Phi}^2}$  or  $\mathcal{T}^{\tilde{\Phi}^3}$  from realizations of  $\underline{\mathbf{z}}$  and (16)-(17) or (18)-(19), respectively.

Tensor entries are computed one by one just like in the real case. We call  $\tilde{\Gamma}_z$  the first generating function of  $\underline{\mathbf{z}}$  defined in  $\mathbb{R}^{2N}$  by:

$$\tilde{\Gamma}_z(\mathbf{w}^{(s)}) \stackrel{\text{def}}{=} \mathbb{E} \left[ \exp(\mathbf{u}^{(s)T} \mathbf{x} + \mathbf{v}^{(s)T} \mathbf{y}) \right], \quad (20)$$

in order that  $\tilde{\Phi}_z = \log \tilde{\Gamma}_z$ . In practice, the expected value is estimated by the sample mean over all realizations. Note that this estimator is consistent but it leads to a biased estimation of the partial derivatives of  $\tilde{\Phi}_z$ , if the latter are computed by finite differences of (20). As in [30], it is preferred to compute formal derivatives, and estimate the obtained expressions with the help of sample means.

Let us define  $D^*(\mathbf{w}^{(s)})$  as the partial derivatives of  $\tilde{\Gamma}_z(\mathbf{w}^{(s)})$  with respect to the components of vectors  $\mathbf{u}^{(s)}$  and  $\mathbf{v}^{(s)}$ . For instance,  $D_{pq}^{vu}(\mathbf{w}^{(s)})$  is the second-order derivative with respect to component  $p$  of  $\mathbf{v}^{(s)}$  and component  $q$  of  $\mathbf{u}^{(s)}$ . Examples of first, second and third order derivatives are:

$$D_p^u(\mathbf{w}^{(s)}) \stackrel{\text{def}}{=} \frac{\partial \tilde{\Gamma}_z(\mathbf{w}^{(s)})}{\partial u_p} = x_p \tilde{\Gamma}_z(\mathbf{w}^{(s)}),$$

$$D_{pq}^{vu}(\mathbf{w}^{(s)}) \stackrel{\text{def}}{=} \frac{\partial^2 \tilde{\Gamma}_z(\mathbf{w}^{(s)})}{\partial v_p \partial u_q} = y_p x_q \tilde{\Gamma}_z(\mathbf{w}^{(s)}),$$

$$D_{pqr}^{vuu}(\mathbf{w}^{(s)}) \stackrel{\text{def}}{=} \frac{\partial^3 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p \partial v_q \partial u_r} = y_p y_q x_r \tilde{\Gamma}_z(\mathbf{w}^{(s)}).$$

The first order derivatives of  $\tilde{\Phi}_z$  are given by:

$$\frac{\partial \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p} = \frac{D_p^u(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z(\mathbf{w}^{(s)})}; \quad \frac{\partial \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial v_p} = \frac{D_p^v(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z(\mathbf{w}^{(s)})}. \quad (21)$$

At order 2, the elements of  $\mathcal{T}^{\tilde{\Phi}^2}$  (i.e. second order derivatives) are obtained by differentiating (21):

$$\mathcal{T}_{pq s1}^{\tilde{\Phi}^2} = \frac{D_{pq}^{uu}(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z(\mathbf{w}^{(s)})} - \frac{D_p^u(\mathbf{w}^{(s)})D_q^u(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z^2(\mathbf{w}^{(s)})}, \quad (22)$$

$$\mathcal{T}_{pq s2}^{\tilde{\Phi}^2} = \frac{D_{pq}^{vv}(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z(\mathbf{w}^{(s)})} - \frac{D_p^v(\mathbf{w}^{(s)})D_q^v(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z^2(\mathbf{w}^{(s)})}, \quad (23)$$

$$\mathcal{T}_{pq s3}^{\tilde{\Phi}^2} = \frac{D_{pq}^{uv}(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z(\mathbf{w}^{(s)})} - \frac{D_p^u(\mathbf{w}^{(s)})D_q^v(\mathbf{w}^{(s)})}{\tilde{\Gamma}_z^2(\mathbf{w}^{(s)})}. \quad (24)$$

At order 3 the elements of  $\mathcal{T}^{\tilde{\Phi}^3}$  (i.e. third order derivatives) are obtained by differentiating order 2 equations:

$$\begin{aligned} \mathcal{T}_{pqr s1}^{\tilde{\Phi}^3} &= \frac{D_{pqr}^{uuu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z(\mathbf{w}^{(s)})} - \frac{2D_p^u(\mathbf{w}^{(s)})D_q^u(\mathbf{w}^{(s)})D_r^u(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^3(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_r^u(\mathbf{w}^{(s)})D_{pq}^{uu}(\mathbf{w}^{(s)}) + D_p^u(\mathbf{w}^{(s)})D_{qr}^{uu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_q^u(\mathbf{w}^{(s)})D_{pr}^{uu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})}, \end{aligned} \quad (25)$$

$$\begin{aligned} \mathcal{T}_{pqr s2}^{\tilde{\Phi}^3} &= \frac{D_{pqr}^{uvv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z(\mathbf{w}^{(s)})} - \frac{2D_p^u(\mathbf{w}^{(s)})D_q^v(\mathbf{w}^{(s)})D_r^v(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^3(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_r^v(\mathbf{w}^{(s)})D_{pq}^{uv}(\mathbf{w}^{(s)}) + D_p^u(\mathbf{w}^{(s)})D_{qr}^{uv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_q^v(\mathbf{w}^{(s)})D_{pr}^{uv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})}, \end{aligned} \quad (26)$$

$$\begin{aligned} \mathcal{T}_{pqr s3}^{\tilde{\Phi}^3} &= \frac{D_{pqr}^{vvv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z(\mathbf{w}^{(s)})} - \frac{2D_p^v(\mathbf{w}^{(s)})D_q^v(\mathbf{w}^{(s)})D_r^v(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^3(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_r^v(\mathbf{w}^{(s)})D_{pq}^{vv}(\mathbf{w}^{(s)}) + D_p^v(\mathbf{w}^{(s)})D_{qr}^{vv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_q^v(\mathbf{w}^{(s)})D_{pr}^{vv}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})}, \end{aligned} \quad (27)$$

$$\begin{aligned} \mathcal{T}_{pqr s4}^{\tilde{\Phi}^3} &= \frac{D_{pqr}^{vuu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z(\mathbf{w}^{(s)})} - \frac{2D_p^v(\mathbf{w}^{(s)})D_q^v(\mathbf{w}^{(s)})D_r^u(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^3(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_r^u(\mathbf{w}^{(s)})D_{pq}^{vv}(\mathbf{w}^{(s)}) + D_p^v(\mathbf{w}^{(s)})D_{qr}^{vu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})} \\ &\quad - \frac{D_q^v(\mathbf{w}^{(s)})D_{pr}^{vu}(\mathbf{w}^{(s)})}{\tilde{\Phi}_z^2(\mathbf{w}^{(s)})}. \end{aligned} \quad (28)$$

### B. Description of the algorithm

The proposed algorithm is named LEMACAF-C-O, where ‘‘O’’ indicates the order of differentiation. For instance LEMACAF-C-2 consists of iteratively fitting the tensor  $\tilde{\mathcal{T}}^{\tilde{\Phi}^2}$  built from model equations (9)-(11) to  $\mathcal{T}^{\tilde{\Phi}^2}$  built from (22)-(24), using the Levenberg-Marquardt method.

The basic scheme of LEMACAF-C is thus similar to the LEMECAF one. However, in this case, we are dealing with highly structured fourth- and fifth-order tensors (when considering  $\mathcal{T}^{\tilde{\Phi}^2}$  and  $\mathcal{T}^{\tilde{\Phi}^3}$  respectively). Thereby the parameter vector, the cost function and the construction of the Jacobian matrix and the gradient vector used at each LM update are completely different, involving more complicated calculations. In the LEMACAF-C-2 case, the quadratic cost function is defined as:

$$f_{\tilde{\Phi}}(\mathbf{p}) = \frac{1}{2} \|\mathbf{e}_{\tilde{\Phi}}(\mathbf{p})\|_F^2 = \frac{1}{2} \mathbf{e}_{\tilde{\Phi}}^T(\mathbf{p}) \mathbf{e}_{\tilde{\Phi}}(\mathbf{p}),$$

where  $\mathbf{e}_{\tilde{\Phi}}(\mathbf{p}) = \text{vec}(\mathcal{T} - \hat{\mathcal{T}}^{\tilde{\Phi}^2}(\mathbf{p}))$  is the residue and  $\mathbf{p}$  is the parameter vector:

$$\mathbf{p} = \begin{bmatrix} \text{vec}(\hat{\mathbf{A}}^T) \\ \text{vec}(\hat{\mathbf{B}}^T) \\ \text{vec}(\hat{\mathbf{G}}^{11T}) \\ \text{vec}(\hat{\mathbf{G}}^{12T}) \\ \text{vec}(\hat{\mathbf{G}}^{22T}) \end{bmatrix} \in \mathbb{R}^{(2N+3S)K \times 1},$$

The LM update at iteration  $k+1$  is still given by:

$$\mathbf{p}(k+1) = \mathbf{p}(k) - [\mathbf{J}^H(k)\mathbf{J}(k) + \lambda(k)\mathbf{I}]^{-1} \mathbf{g}(k),$$

where  $\mathbf{J}$  and  $\mathbf{g}$  denotes the Jacobian matrix and gradient vectors respectively. These are obtained by computing analytically  $J_{ij}(k) = \frac{\partial e_{\tilde{\Phi}}^i(\mathbf{p})}{\partial p_j}$  and  $\mathbf{g}(k) = \mathbf{J}(k)^T \mathbf{e}_{\tilde{\Phi}}(\mathbf{p})$ .

Elements of the Jacobian matrix are given in Appendix B for LEMACAF-C-2 and LEMACAF-C-3. At every iteration  $k$ ,  $\mathbf{g}$ ,  $\mathbf{J}$ ,  $\mathbf{p}$  and  $\lambda$  are updated according to the LEMACAF scheme, described in section II. After convergence of the algorithm, an estimate  $\hat{\mathbf{H}}$  of the mixture is obtained by  $\hat{\mathbf{H}} = \text{unvec}\{\mathbf{p}_{\hat{\mathbf{A}}} + j\mathbf{p}_{\hat{\mathbf{B}}}\}$  (up to column permutation and scaling).

### V. SIMULATION RESULTS PART I: THE REAL CASE

In this section, we compare both ALESCAF and LEMACAF algorithms on mixtures of synthesized BPSK sources, with the well known 6-BIOME (Blind Identification of Overcomplete MixturEs) algorithm [27], also referred to as ‘‘BIRTH’’ (Blind Identification of mixtures of sources using Redundancies in the daTa Hexacovariance matrix) when the chosen cumulant order is 6. For these comparisons we retained third and fourth order versions of ALESCAF and LEMACAF (respectively: ALESCAF-3, ALESCAF-4, LEMACAF-3 and LEMACAF-4). Note that as far as we know, ALESCAF-4 had never been implemented before. Algorithms are evaluated according to the Normalized Mean Square estimation Error (NMSE):

$$f_H(\mathbf{H}, \hat{\mathbf{H}}) = \frac{\text{vec}(\mathbf{H} - \hat{\mathbf{H}})^T \text{vec}(\mathbf{H} - \hat{\mathbf{H}})}{\text{vec}(\mathbf{H})^T \text{vec}(\mathbf{H})},$$

where the permutation and scaling ambiguities present in  $\hat{\mathbf{H}}$  are fixed in the same manner as in [20].

The precision of the estimation relies upon three main global parameters: the ‘‘underdeterminacy level’’, the number of observations and the signal to noise ratio (SNR). Their respective influences on the five algorithms are evaluated using

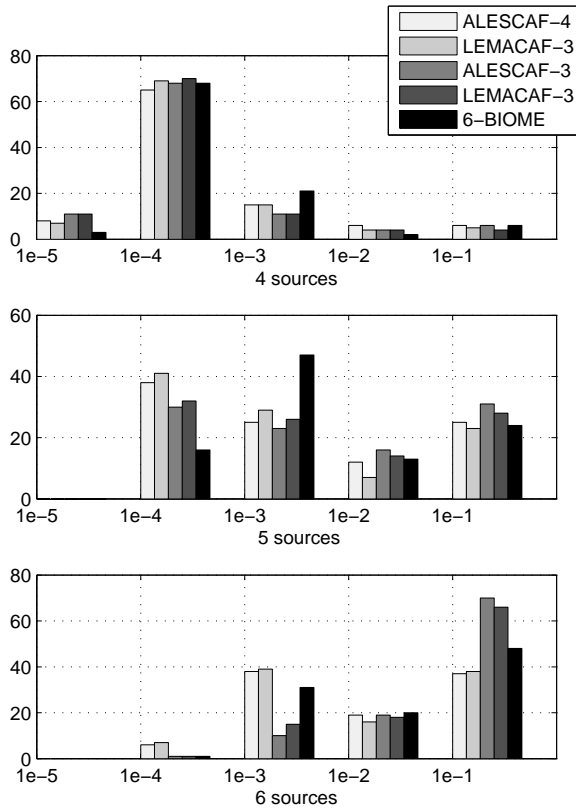


Fig. 1. Real case, NMSE distribution according to the number of sources.

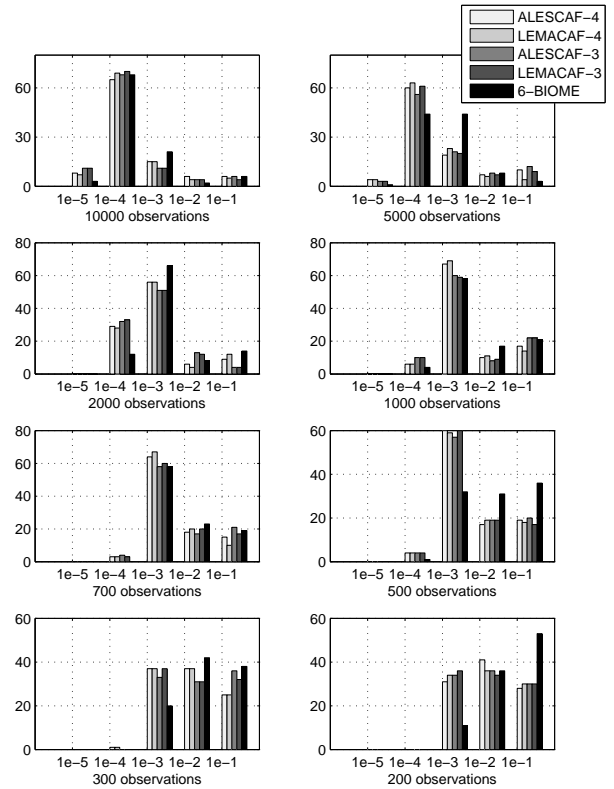


Fig. 2. Real case, NMSE distribution according to the number of observations

distinct Monte-Carlo simulations. Hence, For each simulation and algorithm, NMSE values are computed from 100 independent realizations of source and mixing matrices. The number of sensors,  $N$ , is always equal to 3. Concerning the CAF algorithms, the derivatives are computed at 8 different points (that is,  $R = 8$ ), randomly drawn in the range  $[-1; 1]^N$ .

*A. Impact of the number of sources*

In this first simulation the noise is null, we use 10000 observations and we made three Monte-Carlo experiments corresponding to three different source numbers ( $K = 4$ ,  $K = 5$ ,  $K = 6$ ). Histograms of the identification error are represented in Fig. 1, for the five algorithms, and summarized in table I.

*a) 4 sources:* Whatever the algorithm most NMSE values range between  $10^{-4}$  and  $10^{-2}$  and the largest part is between  $10^{-4}$  and  $10^{-3}$ , indicating that all algorithms perform well. There is no significant difference between CAF algorithms which all performed slightly better than 6-BIOME.

*b) 5 sources:* All NMSE values are greater than  $10^{-4}$ . ALESCAF-4 and LEMACAF-4 provide the best results (40% of NMSE values are less than  $10^{-3}$ ) followed by the order three CAF algorithms (30%) and lastly 6-BIOME (less 20%). According to the median error values, LEMACAF seems more efficient than ALESCAF.

*c) 6 sources:* A majority of Monte-Carlo runs did not converge. This holds true for all algorithms. Only the CAF

algorithms provide some (less than 10%) NMSE values under  $10^{-3}$ . In this case, with about 45% of NMSE values under  $10^{-2}$ , order 4 algorithms are clearly better than order 3 (15%). 6-BIOME is on the average with more than 30%. Once again LEMACAF is slightly better than ALESCAF.

*d) Conclusion:* In this first experiment, it clearly appears that order 4 algorithms are particularly attractive when the underdeterminacy level is high, at the opposite of order 3 algorithms, while 6-BIOME has average results. It also appears that LEMACAF usually provides a better convergence than ALESCAF.

*B. Impact of the number of observations*

Now, the number of sources is set to 4, and the number of observations is varied from 200 to 10000. The results are shown in Fig. 2 in the form of histograms. The variation of the median value of the NMSE is plotted in Fig. 3.

As expected, the performance improves linearly with an increase in the number of observations. Globally, beyond 5000 observations the majority of NMSE values are lower than  $10^{-4}$ . Between 5000 and 1000 observations the NMSE values are comprised between  $10^{-4}$  and  $10^{-2}$ . We have observed no significant differences among the performance of the four CAF algorithms. On the other hand, according to the median plot of Fig. 2, they all give better results than 6-BIOME in all the eight situations. Additionally, these results also show that

TABLE I  
REAL CASE, MINIMAL, MAXIMAL AND MEDIAN VALUE OF THE NMSE ACCORDING TO THE NUMBER OF SOURCES COMPUTED FROM 100 MONTE-CARLO RUNS.

	4 sources			5 sources			6 sources		
	min (1e-4)	max	med (1e-3)	min (1e-3)	max	med (1e-2)	min (1e-3)	max	med (1e-1)
ALESCAF4	0.22	1.04	0.33	0.11	0.89	0.24	0.3	1.43	0.3
LEMACAF4	0.22	1.04	0.27	0.11	1.18	0.16	0.3	1.45	0.2
ALESCAF3	0.26	1.54	0.41	0.14	1.77	0.51	1	1.75	5.38
LEMACAF3	0.26	1.54	0.4	0.14	1.58	0.39	1	1.95	4.91
6-BIOME	0.21	2.08	0.56	0.22	1.93	0.33	0.6	1.98	0.84

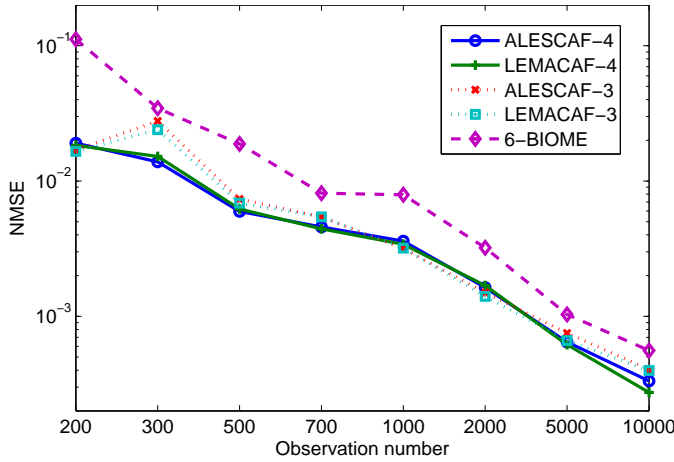


Fig. 3. Real case, evolution of the NMSE median value according to the numbers of observations .

the difference between the two classes of algorithms is more significant under 700 observations.

C. Impact of the signal to noise ratio

Finally, the algorithms are compared with the help of six Monte-Carlo experiments corresponding to different SNRs, and considering 4 sources and 10000 observations. The results are presented in Fig. 4 in an histogram form, while the evolution of the median NMSE value is plotted on Fig. 5.

From an infinite SNR until 20 dB, a large majority (between 60 and 70%) of the NMSE values are comprised between  $10^{-4}$  and  $10^{-3}$ . In the 5 dB case this majority is comprised between  $10^{-3}$  and  $10^{-2}$  and between  $10^{-2}$  et  $10^{-1}$  for the 0 dB case. As can be seen in Fig. 5, CAF results are again very close to each other. CAF algorithms should be preferred for higher SNRs (above 20 dB) whereas 6-BIOME outperforms them at 0 dB SNR. Note that, in this case, very few Monte-Carlo results are satisfactory whatever the algorithm. In the middle of the SNRs range ( i.e. around 5 dB), LEMACAF4, ALESCAF4 and 6-BIOME present similar performances . Finally, LEMACAF is once again slightly better than ALESCAF.

D. Discussion

From the previous simulation results, it can be concluded that the CAF algorithms perform similarly in more "favourable" scenarios (i.e. when the SNR and the number of observations are high and the undeterminacy level is low). However a clear distinction between third- and fourth-order algorithms can be made in more difficult cases. As expected, the

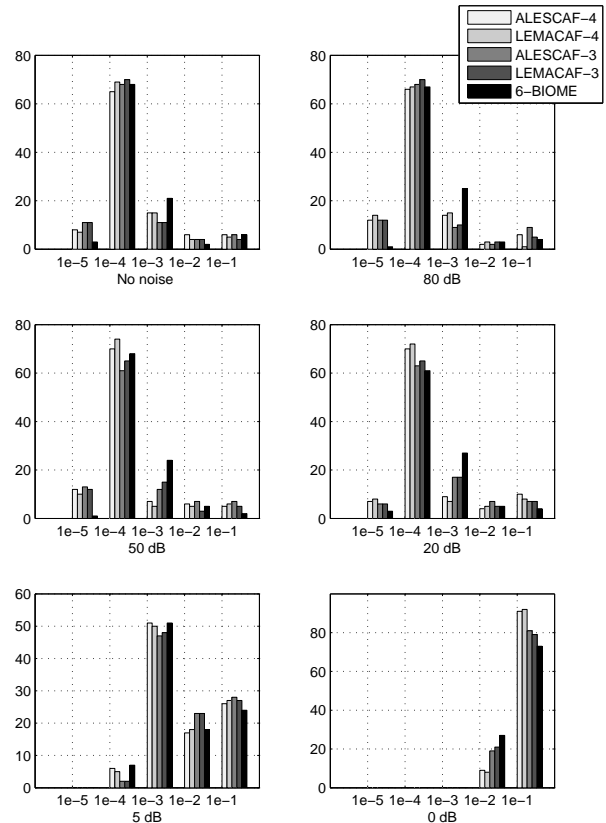


Fig. 4. Real case, NMSE distribution according to the SNR.

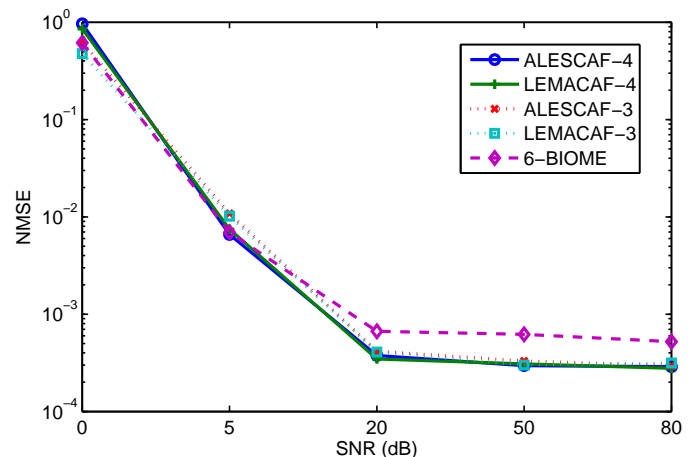


Fig. 5. Real case, Evolution of the median NMSE value according to the SNR

improvement obtained with higher order algorithms becomes significant when the indeterminacy level increases. These results show that LEMACAF is an interesting alternative to ALESCAF. It is worth mentioning that LEMACAF converges faster than ALESCAF at the price of a higher computational complexity per iteration [20]. Although ALESCAF can still be useful with larger tensors, we think that LEMACAF should be recommended in more practical situations.

By properly choosing the differentiation order, the family of CAF algorithms arises as a better option than the 6-BIOME algorithm, especially when the number of observations is low and/or when the underdeterminacy level is high and/or when the SNR is above 5 dB. Indeed, in many situations, the 6-BIOME algorithm offers similar performance as third-order CAF algorithms however it is consistently overpass by fourth-order CAF algorithm.

VI. SIMULATION RESULTS PART II: THE COMPLEX CASE

In this section, we compare the performances of LEMACAF-C-3, LEMACAF-C-4 and 6-BIOME algorithms considering complex mixtures of synthesized 4-QAM sources at different SNRs and underdeterminacy levels. Similarly to the real case, the different algorithms are evaluated in terms of the NMSE. The influence of the SNR and underdeterminacy level is evaluated with the help of two distinct Monte-Carlo experiments. For each simulation and algorithm, the MSE values are computed from about 50 realizations of sources and mixing matrix. The number of sensors ( $N$ ) is always equal to 3 and the number of observations is set to 20000. At each run, the derivatives used in the LEMACAF-C algorithms are computed at 10 different points ( $R = 10$ ), the real and imaginary parts being randomly drawn in the range  $[-1; 1]^N$ .

A. Impact of the number of sources

We have made three different experiments corresponding to 4, 5 and 6 sources. The SNR is set to 20 dB. An histogram of the NMSE values is given in Fig. 6. Minimal, maximal and median value of the NMSE according to the number of source are given in table II.

e) 4 sources: Fig. 6 clearly shows that both LEMACAF-C algorithms provide statistically the best estimations, since about 40% of the obtained NMSE values are under  $10^{-3}$  against about only 16% for 6-BIOME. Note also that LEMACAF-C-4 performs better than LEMACAF-C-3: 18% of its NMSE values are under  $10^{-3}$  against 8% for LEMACAF-C-3 besides only 8% are above  $5 \cdot 10^{-3}$  against 14% for LEMACAF-C-3 (and 20% for 6-BIOME). Table II confirms this observation. Note that the maximum value obtained with LEMACAF-C-4 is only 0.01.

f) 5 sources: For the three algorithms, there are very few values under  $10^{-3}$  whereas a significant number of NMSE values are between  $10^{-3}$  and  $5 \cdot 10^{-3}$ . LEMACAF-C-4 dominates with 50% of its NMSE values in this range, against 40% for the other algorithms. More generally, in this case, LEMACAF-C-3 and 6-BIOME are slightly equivalent. This is confirmed by the median values stored in Table II.

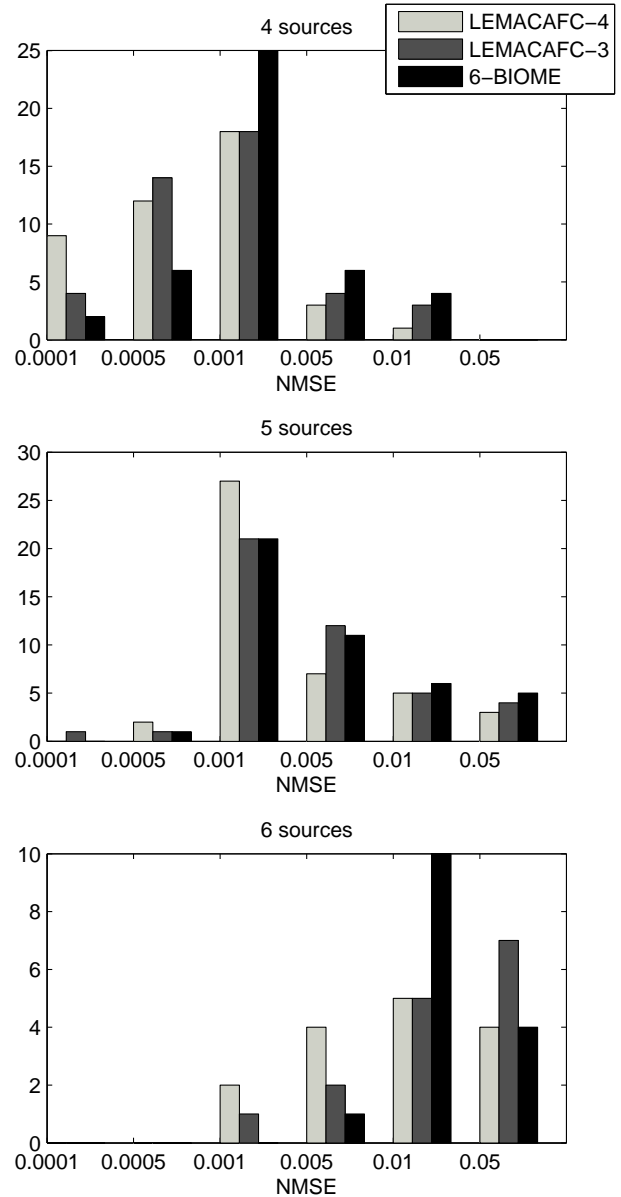


Fig. 6. Complex case, NMSE distribution according to the number of sources.

g) 6 sources: 15 Monte-Carlo runs have been used for this test. This is a difficult situation where the underdeterminacy level is high. Note that the NMSE values of 6-BIOME do not decrease under  $5 \cdot 10^{-3}$ , contrary to LEMACAF-C-4 and LEMACAF-C-3. In addition 6-BIOME provides only one values under  $5 \cdot 10^{-3}$  against 3 for LEMACAF-C-3 and 6 for LEMACAF-C-4. However, according to the median values, the 6-BIOME algorithm offers a performance that is in between third- and fourth-order CAF algorithms.

TABLE II  
 COMPLEX CASE, MINIMAL, MAXIMAL AND MEDIAN VALUE OF THE NMSE ACCORDING TO THE NUMBER OF SOURCES, COMPUTED FROM SEVERAL MONTE-CARLO RUNS.

	4 sources			5 sources			6 sources		
	min (1e-4)	max	med (1e-3)	min (1e-4)	max	med (1e-3)	min (1e-3)	max	med (1e-2)
LEMACAFC4	2.9	0.01	1	9.1	0.13	3.2	3.6	0.32	1.2
LEMACAFC3	4.0	0.03	1	3.0	0.83	4.5	4.0	0.36	4.1
6-BIOME	3.6	0.02	2	5.3	1.63	4.9	7.1	0.14	2.2

B. Impact of the signal to noise ratio

In the following simulations, the algorithms are compared for five SNR values (5 dB, 10 dB, 20 dB, 30 dB and 50 dB), and assuming 4 sources. The results are depicted in Fig. 7 and 8. Regarding the median NMSE plotted in Fig. 8, it can be observed that performances logically degrade as the SNR decreases. However, one can clearly discriminate two opposite situations. Above 5dB SNR, LEMACAFC-3 and LEMACAFC-4 offer similar performances and clearly outperform the 6-BIOME algorithm. On the other hand, at around 5dB SNR, 6-BIOME provides better results.

The histograms plotted in Fig. 7 refine these observations. In the SNR range from 50 dB to 10 dB, the number of NMSE values above  $5 \cdot 10^{-3}$  remains stable while a large majority of NMSE values is consistently located between  $10^{-3}$  and  $5 \cdot 10^{-3}$ . This means that all algorithms perform satisfactorily in these conditions.

Now, let us focus on the number of NMSE values located under  $10^{-3}$ . This number gradually decreases with the SNR, from about 50% to 12% for LEMACAFC-4, 40% to 18% for LEMACAFC-3 and 20% to 4% for 6-BIOME, indicating the superiority of the CAF approach in this range. On the other hand, in the 5 dB SNR case, 25% of 6-BIOME NMSE values are less than  $5 \cdot 10^{-3}$  against about 8 % and 4 % for LEMACAFC-4 and LEMACAFC-3 respectively. In other words, it confirms our previous observations, since for SNR values which are higher than 5 dB, CAF algorithms are more attractive than 6-BIOME, otherwise the latter is slightly more efficient. Furthermore, it is worth noting that at the exception of the 10 dB SNR case, LEMACAFC-4 consistently provides a larger number of NMSE values located under  $10^{-3}$  than LEMACAFC-3 does. Therefore, higher order version of LEMACAFC still promises a better convergence (from a statistical point view).

VII. CONCLUSION

In this paper we have proposed several new contributions to the blind identification of underdetermined mixtures using the characteristic function. Notably, we have extended the theory of CAF approach to the case of complex mixtures of complex sources. Hence an appropriate core equation was developed to cope with the complex case. By differentiating this core equation, we obtained a generalized tensor decomposition from which an estimation of the mixing matrix can be deduced. A Levenberg-Marquardt algorithm, herein called LEMACAFC, was proposed to blindly estimate the mixing matrix by exploiting the structure this tensor decomposition. Two performance studies were carried out. In a first part, a new LEMACAF algorithm, suitable for the real case, was compared

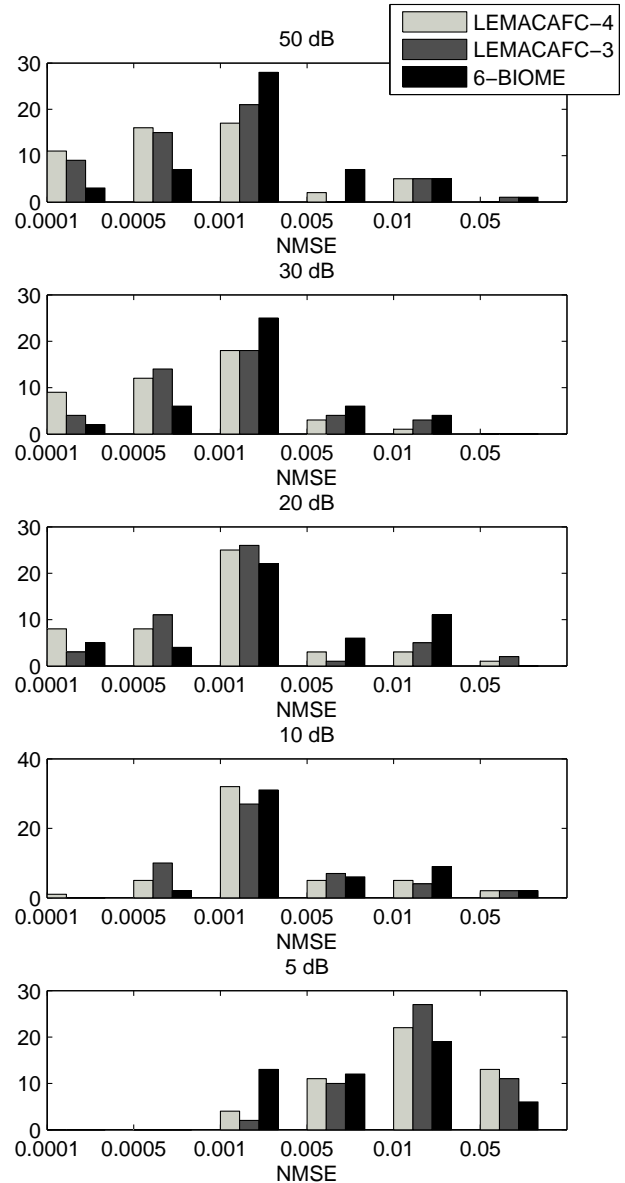


Fig. 7. Complex case, distribution according to the SNR.

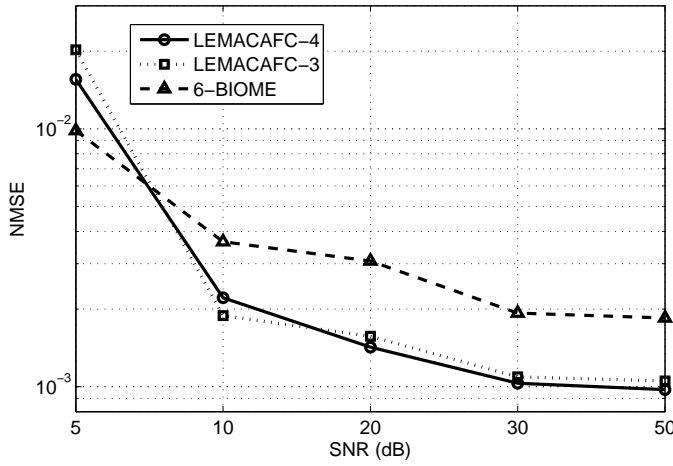


Fig. 8. Complex case, evolution of the NMSE median value according to the SNR.

to higher order versions of ALESCAF in various situations involving BPSK sources. In a second part, the performance of the LEMACAFc algorithm using third- and fourth-order derivatives were evaluated from mixtures of 4-QAM sources. In each situation, the 6-BIOME algorithm was used as a reference for the comparison.

From our simulation results, in both the real and complex cases, we can recommend the use of the CAF family of algorithms, especially in the case of highly underdetermined mixtures and the medium-to-high signal to noise ratios (say, above 5 dB). The choice of the differentiation order is led by the desired tradeoff between estimation accuracy and computational cost. In the least favorable situation, we recommend the use of higher order derivatives. In practice it appears that order 3 versions are enough to compete with (and often outperform) an algorithm using 6-order statistics such as 6-BIOME.

In the case of real sources, although LEMACAFc is applicable, one should prefer ALESCAF or LEMACAF algorithms since they are simpler and quicker. Since both offer quite similar performance in terms of estimation error, the choice is led by the convergence speed, which means that ALESCAF is only preferable for large tensors.

Higher order versions of LEMACAFc, such as LEMACAFc-4 algorithm are time consuming. This is mainly due to the computation of the Jacobian matrix. Our results show that there is generally no need to go beyond order 4, which would be very computationally costly (the computational complexity dramatically increases with the differentiation order). Although we have focused our attention on simplest derivatives, other derivative tensors could have been constructed for higher differentiation orders, leading to different versions of the LEMACAFc algorithm. For example, one can combine derivatives of different differentiation orders to increase diversity, as demonstrated in [30] in the real case.

Finally, as we have pointed out previously, the higher-order derivative tensors that arise when working with complex mixtures of complex sources follow a decomposition in a sum of structured CanD blocks. We think that block decompositions [37] can be used to decompose these tensors. In this context,

it would be interesting to study uniqueness issues and their practical implications to the CAF-based BI problem.

## APPENDIX A

### COMPUTATIONAL DETAILS OF FIRST AND SECOND ORDER DERIVATIVES OF $\tilde{\Phi}_z(\Re\{\mathbf{w}\}, \Im\{\mathbf{w}\})$

The differentiation of (8) with respect to  $u_p$ ,  $p = 1 \dots N$  gives:

$$\begin{aligned} \frac{\partial \tilde{\Phi}_z(\mathbf{w})}{\partial u_p} &= \sum_{k=1}^K \frac{\partial \tilde{\varphi}_k(g(\mathbf{w}))}{\partial u_p}, \\ &= \sum_{k=1}^K \left( \frac{\partial \tilde{\varphi}_k(g)}{\partial g_1} \frac{\partial g_1(\mathbf{w})}{\partial u_p} + \frac{\partial \tilde{\varphi}_k(g)}{\partial g_2} \frac{\partial g_2(\mathbf{w})}{\partial u_p} \right), \\ &= \sum_{k=1}^K \left( \frac{\partial \tilde{\varphi}_k(g)}{\partial g_1} A_{pk} - \frac{\partial \tilde{\varphi}_k(g)}{\partial g_2} B_{pk} \right). \end{aligned} \quad (29)$$

In the same way, the differentiation with respect to  $v_p$  gives:

$$\frac{\partial \tilde{\Phi}_z(\mathbf{w})}{\partial v_p} = \sum_{k=1}^K \left( \frac{\partial \tilde{\varphi}_k(g)}{\partial g_1} B_{pk} + \frac{\partial \tilde{\varphi}_k(g)}{\partial g_2} A_{pk} \right). \quad (30)$$

Now we differentiate (29) and (30) with respect to  $u_q^{(s)}$  and  $v_q^{(s)}$ ,  $q = 1 \dots N$ .

$$\begin{aligned} \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial u_q} &= \sum_{k=1}^K \left( \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_1 \partial u_q} A_{pk} - \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_2 \partial u_q} B_{pk} \right), \\ &= \frac{\partial}{\partial g_1} \sum_{k=1}^K \frac{\partial \tilde{\varphi}_k(g)}{\partial u_q} A_{pk} - \frac{\partial}{\partial g_2} \sum_{k=1}^K \frac{\partial \tilde{\varphi}_k(g)}{\partial u_q} B_{pk}. \end{aligned} \quad (31)$$

Substituting (29) into (31) yields:

$$\begin{aligned} \frac{\partial^2 \tilde{\Phi}_z(\mathbf{w}^{(s)})}{\partial u_p \partial u_q} &= \sum_{k=1}^K A_{pk} \left( \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_1 \partial g_1} A_{qk} - \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_1 \partial g_2} B_{qk} \right) \\ &\quad - \sum_{k=1}^K B_{pk} \left( \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_2 \partial g_1} A_{qk} - \frac{\partial^2 \tilde{\varphi}_k(g)}{\partial g_2 \partial g_2} B_{qk} \right). \end{aligned} \quad (32)$$

Finally, using the fact that  $G_{sk}^{12} = G_{sk}^{21}$ , we obtain (9). The same reasoning can be applied to obtain the successive derivatives of (30) and (29) with respect to  $v_p$  and  $v_q$ , hence (10) and (11).

## APPENDIX B

### JACOBIAN FORMULATION FOR LEMACAFc-2 AND LEMACAFc-3

We provide the details of the structure of the Jacobian matrix used in LEMACAFc-2 and LEMACAFc-3, (higher orders can be obtained in a similar manner). Although the Jacobian can be computed automatically for a given differentiation order, this solution is usually very time consuming. Therefore, one would rather use the following pre-calculated entries. The gradient can be directly deduced from the Jacobian and the error term.

### A. Jacobian for LEMACAF-C-2

$e$  is the multiway error function which is defined as follows:

$$e_{pqst}(\mathbf{A}, \mathbf{B}, \mathbf{G}^{11}, \mathbf{G}^{12}, \mathbf{G}^{22}) \stackrel{\text{def}}{=} \hat{\mathcal{T}}_{pqst}^{\tilde{\Phi}2}(\mathbf{A}, \mathbf{B}, \mathbf{G}^{11}, \mathbf{G}^{12}, \mathbf{G}^{22}) - \mathcal{T}_{pqst}^{\tilde{\Phi}2}$$

The Jacobian matrix  $\mathbf{J}$  of  $e$  is built as a  $3 \times 5$  blocks matrix:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}^{A_1} & \mathbf{J}^{B_1} & \mathbf{J}^{G_1^{11}} & \mathbf{J}^{G_1^{12}} & \mathbf{J}^{G_1^{22}} \\ \mathbf{J}^{A_2} & \mathbf{J}^{B_2} & \mathbf{J}^{G_2^{11}} & \mathbf{J}^{G_2^{12}} & \mathbf{J}^{G_2^{22}} \\ \mathbf{J}^{A_3} & \mathbf{J}^{B_3} & \mathbf{J}^{G_3^{11}} & \mathbf{J}^{G_3^{12}} & \mathbf{J}^{G_3^{22}} \end{bmatrix}$$

$\mathbf{J}^{A_t}, \mathbf{J}^{B_t}, \mathbf{J}^{G_t^{11}}, \mathbf{J}^{G_t^{12}}, \mathbf{J}^{G_t^{22}}$  contain partial derivatives of  $e_{\dots t}$  with respects to the elements of  $A, B, G^{11}, G^{12}$  and  $G^{22}$  respectively. For instance, defining:

$$i = (p-1)SN + (q-1)N + s,$$

$$j_1 = (n-1)N + k,$$

$$j_2 = (l-1)S + k$$

with  $p = 1 \dots N, q = 1 \dots N, n = 1 \dots N, s = 1 \dots S$  and  $k = 1 \dots K$  we have:

$$\begin{aligned} \mathbf{J}_{ij_1}^{A_2} &\stackrel{\text{def}}{=} \frac{\partial e_{pqst}(\mathbf{A}, \mathbf{B}, \mathbf{G}^{11}, \mathbf{G}^{12}, \mathbf{G}^{22})}{\partial A_{nk}} \\ &= \frac{\partial \hat{\mathcal{T}}_{pqst}^{\tilde{\Phi}2}(\mathbf{A}, \mathbf{B}, \mathbf{G}^{11}, \mathbf{G}^{12}, \mathbf{G}^{22})}{\partial A_{nk}} \end{aligned}$$

Each block has its own analytical formulation. Hence,  $\mathbf{J}$  is computed one block after another. Block entries can be computed term by term by differentiation of (9), (10) and (11). One obtains the following relationships (were  $\delta$  is the Kronecker delta function):

$$J_{ij_1}^{A_1} = \delta_{np} (A_{qk} G_{sk}^{11} - B_{qk} G_{sk}^{12}) + \delta_{nq} (A_{pk} G_{sk}^{11} - B_{pk} G_{sk}^{12})$$

$$J_{ij_1}^{A_2} = \delta_{np} (A_{qk} G_{sk}^{22} + B_{qk} G_{sk}^{12}) + \delta_{nq} (B_{pk} G_{sk}^{12} + A_{pk} G_{sk}^{22})$$

$$J_{ij_1}^{A_3} = \delta_{np} (B_{qk} G_{sk}^{11} + A_{qk} G_{sk}^{12}) + \delta_{nq} (A_{pk} G_{sk}^{12} - B_{pk} G_{sk}^{22})$$

$$J_{ij_1}^{B_1} = \delta_{np} (B_{qk} G_{sk}^{22} - A_{qk} G_{sk}^{12}) + \delta_{nq} (B_{pk} G_{sk}^{22} - A_{pk} G_{sk}^{12})$$

$$J_{ij_1}^{B_2} = \delta_{np} (B_{qk} G_{sk}^{11} + A_{qk} G_{sk}^{12}) + \delta_{nq} (B_{pk} G_{sk}^{11} + A_{pk} G_{sk}^{12})$$

$$J_{ij_1}^{B_3} = -\delta_{np} (B_{pk} G_{sk}^{12} + A_{pk} G_{sk}^{22}) + \delta_{nq} (A_{qk} G_{sk}^{11} - B_{qk} G_{sk}^{22})$$

$$J_{ij_2}^{G_1^{11}} = \delta_{ls} (A_{pk} A_{qk})$$

$$J_{ij_2}^{G_2^{11}} = \delta_{ls} (B_{pk} B_{qk})$$

$$J_{ij_2}^{G_3^{11}} = \delta_{ls} (A_{pk} B_{qk})$$

$$J_{ij_2}^{G_1^{12}} = -\delta_{ls} (A_{pk} B_{qk} + B_{pk} A_{qk})$$

$$J_{ij_2}^{G_2^{12}} = \delta_{ls} (B_{pk} A_{qk} + A_{pk} B_{qk})$$

$$J_{ij_2}^{G_3^{12}} = \delta_{ls} (A_{pk} A_{qk} - B_{pk} B_{qk})$$

$$J_{ij_2}^{G_1^{22}} = \delta_{ls} (B_{pk} B_{qk})$$

$$J_{ij_2}^{G_2^{22}} = \delta_{ls} (A_{pk} A_{qk})$$

$$J_{ij_2}^{G_3^{22}} = -\delta_{ls} (B_{pk} A_{qk})$$

### B. Jacobian for LEMACAF-C-3

At order 3, we have now a  $4 \times 6$  blocks matrix:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}^{A_1} & \mathbf{J}^{B_1} & \mathbf{J}^{G_1^{111}} & \mathbf{J}^{G_1^{211}} & \mathbf{J}^{G_1^{221}} & \mathbf{J}^{G_1^{222}} \\ \mathbf{J}^{A_2} & \mathbf{J}^{B_2} & \mathbf{J}^{G_2^{111}} & \mathbf{J}^{G_2^{211}} & \mathbf{J}^{G_2^{221}} & \mathbf{J}^{G_2^{222}} \\ \mathbf{J}^{A_3} & \mathbf{J}^{B_3} & \mathbf{J}^{G_3^{111}} & \mathbf{J}^{G_3^{211}} & \mathbf{J}^{G_3^{221}} & \mathbf{J}^{G_3^{222}} \\ \mathbf{J}^{A_4} & \mathbf{J}^{B_4} & \mathbf{J}^{G_4^{111}} & \mathbf{J}^{G_4^{211}} & \mathbf{J}^{G_4^{221}} & \mathbf{J}^{G_4^{222}} \end{bmatrix}$$

and we define

$$i = (p-1)SN^2 + (q-1)N^2 + (r-1)N + s,$$

$$j_1 = (n-1)N + k,$$

$$j_2 = (l-1)S + k,$$

with  $p = 1 \dots N, q = 1 \dots N, r = 1 \dots N, n = 1 \dots N, s = 1 \dots S$ , and  $k = 1 \dots K$  Block entries are computed by differentiation of (12-15).

$$\begin{aligned} J_{ij_1}^{A_1} &= \delta_{np} (A_{qk} A_{rk} G_{sk}^{111} - A_{qk} B_{rk} G_{sk}^{211} \\ &\quad - B_{qk} A_{rk} G_{sk}^{211} + B_{qk} B_{rk} G_{sk}^{221}) \\ &\quad + \delta_{nq} (A_{pk} A_{rk} G_{sk}^{111} - A_{pk} B_{rk} G_{sk}^{211} \\ &\quad - B_{pk} A_{rk} G_{sk}^{211} + B_{pk} B_{rk} G_{sk}^{221}) \\ &\quad + \delta_{nr} (A_{pk} A_{qk} G_{sk}^{111} - A_{pk} B_{qk} G_{sk}^{211} \\ &\quad - B_{pk} A_{qk} G_{sk}^{211} + B_{pk} B_{qk} G_{sk}^{221}) \end{aligned}$$

$$\begin{aligned} J_{ij_1}^{A_2} &= \delta_{np} (A_{qk} B_{rk} G_{sk}^{111} + A_{qk} A_{rk} G_{sk}^{211} \\ &\quad - B_{qk} B_{rk} G_{sk}^{211} - B_{qk} A_{rk} G_{sk}^{221}) \\ &\quad + \delta_{nq} (A_{pk} B_{rk} G_{sk}^{111} + A_{pk} A_{rk} G_{sk}^{211} \\ &\quad - B_{pk} B_{rk} G_{sk}^{211} - B_{pk} A_{rk} G_{sk}^{221}) \\ &\quad + \delta_{nr} (A_{pk} A_{qk} G_{sk}^{211} - A_{pk} B_{qk} G_{sk}^{221} \\ &\quad - B_{pk} A_{qk} G_{sk}^{221} + B_{pk} B_{qk} G_{sk}^{222}) \end{aligned}$$

$$\begin{aligned} J_{ij_1}^{A_3} &= \delta_{np} (B_{qk} B_{rk} G_{sk}^{211} + B_{qk} A_{rk} G_{sk}^{221} \\ &\quad + A_{qk} B_{rk} G_{sk}^{221} + A_{qk} A_{rk} G_{sk}^{222}) \\ &\quad + \delta_{nq} (B_{pk} B_{rk} G_{sk}^{211} + B_{pk} A_{rk} G_{sk}^{221} \\ &\quad + A_{pk} B_{rk} G_{sk}^{221} + A_{pk} A_{rk} G_{sk}^{222}) \\ &\quad + \delta_{nr} (B_{pk} B_{qk} G_{sk}^{211} + B_{pk} A_{qk} G_{sk}^{221} \\ &\quad + A_{pk} B_{qk} G_{sk}^{221} + A_{pk} A_{qk} G_{sk}^{222}) \end{aligned}$$

$$\begin{aligned} J_{ij_1}^{A_4} &= \delta_{np} (B_{qk} A_{rk} G_{sk}^{211} - B_{qk} B_{rk} G_{sk}^{221} \\ &\quad + A_{qk} A_{rk} G_{sk}^{221} - A_{qk} B_{rk} G_{sk}^{222}) \\ &\quad + \delta_{nq} (B_{pk} A_{rk} G_{sk}^{211} - B_{pk} B_{rk} G_{sk}^{221} \\ &\quad + A_{pk} A_{rk} G_{sk}^{221} - A_{pk} B_{rk} G_{sk}^{222}) \\ &\quad + \delta_{nr} (B_{pk} B_{qk} G_{sk}^{111} + B_{pk} A_{qk} G_{sk}^{211} \\ &\quad + A_{pk} B_{qk} G_{sk}^{211} + A_{pk} A_{qk} G_{sk}^{221}) \end{aligned}$$

## REFERENCES

- $$J_{ij_1}^{B_1} = \delta_{np}(-A_{qk}A_{rk}G_{sk}^{211} + A_{qk}B_{rk}G_{sk}^{221} + B_{qk}A_{rk}G_{sk}^{221} - B_{qk}B_{rk}G_{sk}^{222}) + \delta_{nq}(-A_{pk}A_{rk}G_{sk}^{211} + A_{pk}B_{rk}G_{sk}^{221} + B_{pk}A_{rk}G_{sk}^{221} - B_{pk}B_{rk}G_{sk}^{222}) + \delta_{nr}(-A_{pk}A_{qk}G_{sk}^{211} + A_{pk}B_{qk}G_{sk}^{221} + B_{pk}A_{qk}G_{sk}^{221} - B_{pk}B_{qk}G_{sk}^{222})$$
- $$J_{ij_1}^{B_2} = \delta_{np}(-A_{qk}B_{rk}G_{sk}^{211} - A_{qk}A_{rk}G_{sk}^{221} + B_{qk}B_{rk}G_{sk}^{221} + B_{qk}A_{rk}G_{sk}^{222}) + \delta_{nq}(-A_{pk}B_{rk}G_{sk}^{211} - A_{pk}A_{rk}G_{sk}^{221} + B_{pk}B_{rk}G_{sk}^{221} + B_{pk}A_{rk}G_{sk}^{222}) + \delta_{nr}(A_{pk}A_{qk}G_{sk}^{111} - A_{pk}B_{qk}G_{sk}^{211} - B_{pk}A_{qk}G_{sk}^{211} + B_{pk}B_{qk}G_{sk}^{221})$$
- $$J_{ij_1}^{B_3} = \delta_{np}(B_{qk}B_{rk}G_{sk}^{111} + B_{qk}A_{rk}G_{sk}^{211} + A_{qk}B_{rk}G_{sk}^{211} + A_{qk}A_{rk}G_{sk}^{221}) + \delta_{nq}(B_{pk}B_{rk}G_{sk}^{111} + B_{pk}A_{rk}G_{sk}^{211} + A_{pk}B_{rk}G_{sk}^{211} + A_{pk}A_{rk}G_{sk}^{221}) + \delta_{nr}(B_{pk}B_{qk}G_{sk}^{111} + B_{pk}A_{qk}G_{sk}^{211} + A_{pk}B_{qk}G_{sk}^{211} + A_{pk}A_{qk}G_{sk}^{221})$$
- $$J_{ij_1}^{B_4} = \delta_{np}(B_{qk}A_{rk}G_{sk}^{111} - B_{qk}B_{rk}G_{sk}^{211} + A_{qk}A_{rk}G_{sk}^{211} - A_{qk}B_{rk}G_{sk}^{221}) + \delta_{nq}(B_{pk}A_{rk}G_{sk}^{111} - B_{pk}B_{rk}G_{sk}^{211} + A_{pk}A_{rk}G_{sk}^{211} - A_{pk}B_{rk}G_{sk}^{221}) + \delta_{nr}(-B_{pk}B_{qk}G_{sk}^{211} - B_{pk}A_{qk}G_{sk}^{221} - A_{pk}B_{qk}G_{sk}^{221} - A_{pk}A_{qk}G_{sk}^{222})$$
- $$J_{ij_2}^{G_1^{111}} = \delta_{ls}(A_{pk}A_{qk}A_{rk}) ; J_{ij_2}^{G_2^{222}} = -\delta_{ls}(B_{pk}B_{qk}B_{rk})$$
- $$J_{ij_2}^{G_2^{111}} = \delta_{ls}(A_{pk}A_{qk}B_{rk}) ; J_{ij_2}^{G_2^{222}} = \delta_{ls}(B_{pk}B_{qk}A_{rk})$$
- $$J_{ij_2}^{G_3^{111}} = \delta_{ls}(B_{pk}B_{qk}B_{rk}) ; J_{ij_2}^{G_3^{222}} = \delta_{ls}(A_{pk}A_{qk}A_{rk})$$
- $$J_{ij_2}^{G_4^{111}} = \delta_{ls}(B_{pk}B_{qk}A_{rk}) ; J_{ij_2}^{G_4^{222}} = -\delta_{ls}(A_{pk}A_{qk}B_{rk})$$
- $$J_{ij_2}^{G_1^{211}} = -\delta_{ls}(A_{pk}A_{qk}B_{rk} + A_{pk}B_{qk}A_{rk} + B_{pk}A_{qk}A_{rk})$$
- $$J_{ij_2}^{G_2^{211}} = \delta_{ls}(A_{pk}A_{qk}A_{rk} - A_{pk}B_{qk}B_{rk} - B_{pk}A_{qk}B_{rk})$$
- $$J_{ij_2}^{G_3^{211}} = \delta_{ls}(B_{pk}B_{qk}A_{rk} + B_{pk}A_{qk}B_{rk} + A_{pk}B_{qk}B_{rk})$$
- $$J_{ij_2}^{G_4^{211}} = -\delta_{ls}(B_{pk}B_{qk}B_{rk} - B_{pk}A_{qk}A_{rk} - A_{pk}B_{qk}A_{rk})$$
- $$J_{ij_2}^{G_1^{221}} = \delta_{ls}(A_{pk}B_{qk}B_{rk} + B_{pk}A_{qk}B_{rk} + B_{pk}B_{qk}A_{rk})$$
- $$J_{ij_2}^{G_2^{221}} = -\delta_{ls}(A_{pk}B_{qk}A_{rk} + B_{pk}A_{qk}A_{rk} - B_{pk}B_{qk}B_{rk})$$
- $$J_{ij_2}^{G_3^{221}} = \delta_{ls}(B_{pk}A_{qk}A_{rk} + A_{pk}B_{qk}A_{rk} + A_{pk}A_{qk}B_{rk})$$
- $$J_{ij_2}^{G_4^{221}} = -\delta_{ls}(B_{pk}A_{qk}B_{rk} + A_{pk}B_{qk}B_{rk} - A_{pk}A_{qk}A_{rk})$$
- [1] P. COMON, "Independent Component Analysis," In Higher Order Statistics, J-L. Lacoume, editor, pp. 29–38. Elsevier, Amsterdam, London, 1992.
- [2] C. ESTEVAO and R. FERNANDES and G. FAVIER and J. C. MOTA, "Blind channel identification algorithms based on the PARAFAC decomposition of cumulant tensors: the single and multiuser cases," Signal Processing, 88, 6, pp. 1382–1401, 2008.
- [3] F. ASANO and S. IKEDA and M. OGAWA and H. ASOH and N. KITAWAKI, "Combined approach of array processing and independent component analysis for blind separation of acoustic signals," IEEE Transactions On Speech and Audio Processing, 11, 3, pp. 204–215, 2003.
- [4] A. KACHENOURA and L. ALBERA and L. SENHADJI and P. COMON, "ICA: a potential tool for BCI systems," IEEE Signal Processing Magazine, special issue on Brain-Computer Interfaces, 25, 3, pp. 57–68, 2008.
- [5] L. DE LATHAUWER and D. Callaerts and B. DE MOOR and J. Vandewalle, "Fetal electrocardiogram extraction by source subspace separation," in IEEE Workshop on Higher Order Statistics, Girona, Spain, pp. 134–138, 1995.
- [6] A. CICHOCKI and S.-I. AMARI, Adaptive Blind Signal and Image Processing. New York: Wiley, 2002.
- [7] P. COMON and C. JUTTEN, Handbook of Blind Source Separation, Independent Component Analysis and Applications. Academic Press, 2010, ISBN: 978-0-12-374726-6.
- [8] J. D. CARROLL and J. J. CHANG, "Analysis of Individual Differences in Multidimensional Scaling via N-Way Generalization of Eckart-Young Decomposition," Psychometrika, 35, 3, pp. 283–319, 1970.
- [9] P. BÜRGISSER and M. CLAUSEN and M. A. SHOKROLLAHI, Algebraic Complexity Theory, 315. Springer, 1997.
- [10] R. BRO, "PARAFAC, Tutorial and Applications," Chemom. Intel. Lab. Syst., 38, pp. 149–171, 1997.
- [11] C.A. STEDMON and S. MARKAGER and R. BRO, "Tracing dissolved organic matter in aquatic environments using a new approach to fluorescence spectroscopy," Marine Chemistry, 82, 3-4, pp. 239–254, 2003.
- [12] R.A. HARSHMAN, "Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis," UCLA Working Papers in Phonetics, 16, pp. 1–84, 1970.
- [13] A. SMILDE and R. BRO and P. GELADI "Multi-Way Analysis," Wiley, 2004
- [14] R. A. HARSHMAN, "Determination and proof of minimum uniqueness conditions for PARAFAC-1," UCLA Working Papers in Phonetics, 22, pp. 111–117, 1972.
- [15] J. B. KRUSKAL, "Three-Way Arrays: Rank and Uniqueness of Trilinear Decompositions," Linear Algebra and Applications, 18, 95–138, 1977.
- [16] J.M.F. TEN BERGE and N.D. SIDIROPOULOS, "On uniqueness in CANDECOMP/PARAFAC," Psychometrika, 67, 399–409, 2002.
- [17] T. JIANG and N.D. SIDIROPOULOS, "Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints," Trans. on Sig. Proc., 52, 9 2625–2636, 2004.
- [18] L. DE LATHAUWER, "A Link between Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization," SIAM Journal on Matrix Analysis, 28, 3 642–666, 2006.
- [19] G. TOMASI and R. BRO, "A comparison of algorithms for fitting the parafac model," Comp. Stat. Data Anal., vol. 50, pp. 1700–1734, 2006.
- [20] P. COMON and X. LUCIANI and A.L.F. DE ALMEIDA, "Tensor Decompositions, Alternating Least Squares and other Tales," Journal of Chemometrics, 23, 9, pp. 393–405, Sept. 2009.
- [21] N. D. SIDIROPOULOS and G. B. GIANNAKIS and R. BRO, "Blind PARAFAC Receivers for DS-CDMA Systems," Trans. on Sig. Proc., 48, 3, pp. 810–823, 2000.
- [22] L. DE LATHAUWER and J. CASTAING, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," IEEE Trans. Signal Process., vol. 56, no. 3, pp. 1096–1105, Mar. 2008.
- [23] P. COMON, "Blind identification and source separation in 2x3 underdetermined mixtures," IEEE Trans. Signal Process., pp. 11–22, Jan. 2004.
- [24] A. L. F. de ALMEIDA and G. FAVIER and J. C. M. MOTA, "PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization," Signal Processing, 87, 2, pp. 337–351, 2007.
- [25] J. F. CARDOSO, "Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors," in Proc. ICASSP'91, Toronto, 1991, pp. 3109–3112.

- [26] L. DE LATHAUWER, J. CASTAING, and J.-F. CARDOSO, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 2965–2973, Feb. 2007.
- [27] L. ALBERA, A. FERREOL, P. COMON, and P. CHEVALIER, "Blind identification of overcomplete mixtures of sources (BIOME)," *Lin. Algebra Appl.*, vol. 391, pp. 1–30, Nov. 2004.
- [28] A. TALEB, "An algorithm for the blind identification of  $n$  independent signals with 2 sensors," *ISSPA'01*, Kuala Lumpur, 2001, vol. 1, pp. 5–8.
- [29] P. COMON and M. RAJIH, "Blind identification of complex underdetermined mixtures," *ICA Conference*, Granada, 2004, pp. 105–112.
- [30] P. COMON and M. RAJIH, "Blind identification of under-determined mixtures based on the characteristic function," *Signal Processing*, 86, 9, pp. 2271–2281, 2006.
- [31] A. M. KAGAN and Y. V. LINNIK and C. R. RAO *Characterization Problems in Mathematical Statistics*, Probability and Mathematical Statistics, Wiley, New York, 1973
- [32] W. FELLER *An Introduction to Probability Theory and its Applications*, Wiley, vol. II, 1966
- [33] M. RAJIH, P. COMON and R. HARSHMAN, "Enhanced Line Search : A Novel Method to Accelerate PARAFAC," *SIAM Journal on Matrix Analysis Appl.*, 30(3):1148–1171, Sept. 2008.
- [34] D. NION, "Methodes PARAFAC generalisees pour l'extraction aveugle de sources. Application aux systemes DS-CDMA," Thesis at the Universite de Cergy-Pontoise, 2007.
- [35] A.L.F. DE ALMEIDA, X. LUCIANI, P. COMON, "Blind identification of underdetermined mixtures based on the hexacovariance and higher-order cyclostationarity," *Proc. SSP'09*, Cardiff, 2009, pp. 669–672.
- [36] K. MADSEN and H. B. NIELSEN and O. TINGLEFF, "Methods for Non-Linear Least Squares Problems," *Technical University of Denmark, Informatics and mathematical Modelling*, 2
- [37] L. DE LATHAUWER, "Decompositions of a Higher-Order Tensor in Block Terms-Part II: Definitions and Uniqueness," *SIAM Journal on Matrix Analysis Appl.*, 30(3):1033–1066, Sept. 2008.



**Xavier Luciani** was born in Toulon, France, in 1979. In 2003 he received both the ISEN engineering diploma (College of electronic and numeric engineering) and a Master's Degree in signal processing from the University of Toulon. He received the Ph.D Degree in engineering sciences from the University of Toulon in 2007. From march 2008 to september 2009, he held a postdoctoral position at the I3S Laboratory, Sophia Antipolis, France. He is currently a postdoctoral researcher with the LTSI Laboratory, U642, INSERM and University of Rennes 1 in

France. Its research interests have been in tensor modeling of fluorescence signals, blind source separation and blind identification based on tensor approaches, algorithms for joint diagonalization and tensor analysis and applications to chemometrics and telecommunications.



**André L. F. de Almeida** received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Ceará, Brazil, in 2001 and 2003, respectively, and the double Ph.D. degree in sciences and teleinformatics engineering from the University of Nice, Sophia Antipolis, France, and the Federal University of Ceará, Fortaleza, Brazil, in 2007. In 2002, he was a visiting researcher at Ericsson Research, Stockholm, Sweden. He was a postdoctoral fellow with the I3S Laboratory, CNRS, Sophia Antipolis, France, from January to December

2008. He is currently an assistant professor at the Department of Teleinformatics Engineering of the Federal University of Ceará, Fortaleza, Brazil. He is also a senior researcher at the Wireless Telecom Research Group (GTEL), where he has worked on several funded research projects. His research interests lie in the area of signal processing for communications, including blind identification and signal separation, space-time processing, multiple-antenna techniques, and multicarrier communications. His recent work has focused on the development of tensor models for transceiver design in wireless communication systems.



**Pierre Comon** (M'87 - SM'95 - F'07) graduated in 1982, and received the Doctorate degree in 1985, both from the University of Grenoble, France. He later received the Habilitation to Lead Researches in 1995, from the University of Nice, France. He has been for nearly 13 years in industry, first with Crouzet-Sextant, Valence, France, between 1982 and 1985, and then with Thomson Marconi, Sophia-Antipolis, France, between 1988 and 1997. He spent 1987 with the ISL laboratory, Stanford University, CA. He joined in 1997 the Eurecom

Institute, Sophia-Antipolis, France, and left during the fall of 1998. He is now research director with CNRS since 1998 at laboratory I3S, Sophia-Antipolis, France. His research interests include High-Order Statistics (HOS), Blind Deconvolution and Equalization, Statistical Signal and Array Processing, Tensor decompositions, Multi-Way Factor Analysis and its applications to biomedical end environment.

Dr. Comon was Associate Editor of the *IEEE Transactions on Signal Processing* from 1995 to 1998, and a member of the French National Committee of Scientific Research from 1995 to 2000. He was the coordinator of the European Basic Research Working Group on HOS, *ATHOS*, from 1992 to 1995. Between 1992 and 1998, he was a member of the Technical and Scientific Council of the Thomson Group. Between 2001 and 2004 he acted as launching Associate Editor with the *IEEE Transactions on Circuits and Systems I*, in the area of Blind Techniques. He is currently belongs to the *SPTM IEEE Technical Committee*, and is a member of the editorial boards of the Elsevier journal *Signal Processing*, and the *SIAM Jour. on Matrix Analysis and Appl.*