Blind underdetermined mixture identification by joint canonical decomposition of HO cumulants
Ahmad Karfoul, Laurent Albera, Gwénaël Birot

To cite this version:


HAL Id: hal-00593307
https://hal.archives-ouvertes.fr/hal-00593307
Submitted on 13 May 2011
BLIND UNDERDETERMINED MIXTURE IDENTIFICATION BY JOINT CANONICAL DECOMPOSITION OF HO CUMULANTS

Ahmad KARFOUL(1,2), Student member, IEEE, Laurent ALBERA(1,2), Member, IEEE, Gwénaël BIROT(1,2)

Abstract—A new family of cumulant-based algorithms is proposed in order to blindly identify potentially underdetermined mixtures of statistically independent sources. These algorithms perform a joint CANonical Decomposition (CAND) of several higher order cumulants through a CAND of a 3-way array with special symmetries. These techniques are studied in terms of identifiability, performance and numerical complexity. From a signal processing viewpoint, the proposed methods are shown i) to have a better estimation resolution and ii) to be able to process more sources than the other classical cumulant-based techniques. Secondly, from a numerical analysis viewpoint, we deal with the identifiability and the convergence speed of several 3-way array decomposition procedures such as the symmetric ACDC scheme. We also show how to accelerate the iterative CAND algorithms by using differently the symmetries of the considered 3-way array. Next, from a multi-linear algebra viewpoint the paper aims at giving some insights on the uniqueness of a joint CAND of several hermitian multi-way arrays compared to the CAND of only one array. In addition, this allows us to extend the concept of Virtual Array (VA) to the case of a combination of several VA’s.

Index Terms—Blind Underdetermined Mixture Identification (BUMI), underdetermined mixture, canonical decomposition, PARAFAC, INDSCAL, ICA, BSS.

I. INTRODUCTION

BLIND Underdetermined Mixture Identification (BUMI) is addressed in this paper. More particularly, we focus on noisy static mixtures of statistical independent sources. In such a context, the BUMI problem is defined by:

\[
\text{Problem 1: Given a vector random process } \{x[k]\}_{k \in \mathbb{N}}, \text{ find a } (N \times P) \text{ mixing matrix } A = [a_1, \ldots, a_P] \text{ potentially underdetermined such that:}
\]

\[
\forall k \in \mathbb{N}, \quad x[k] = As[k] + \nu[k] \tag{1}
\]

where \(a_p\) is the \(p\)-th column vector of \(A\), \(s[k] = [s_1[k], \ldots, s_P[k]]^T\) is a random vector whose \(P\) components are statistically independent and where \(\nu[k]\) is a \((N \times 1)\) Gaussian noise vector, independent of \(s[k]\).

The resolution of problem 1 is of a great research interest given its importance in a great variety of signal processing applications, such as blind source separation [3], [23] or estimation of direction of arrivals [17]. Then, many solutions based on the concept of independent component analysis [11], [22] were proposed in order to solve problem 1 when \(A\) is an overdetermined mixture, say when the number \(N\) of sources is greater or equal than the number \(P\) of sources. For instance some methods iteratively maximize the opposite of the mutual information [4], [25], [37] or the negentropy [20, chapter 8]. Other algorithms resort to cumulants, which allow for a measure of statistical independence [36], [46] less natural but easier to compute. In such a way, semi-algebraic methods were defined in order to benefit from the structure of the data cumulants as a function of the mixture. Particularly, some methods [5], [18], [48] exploit Second Order (SO) cumulants only, while others use either both SO and Fourth Order (FO) cumulants [6], [11], [32], [34], [36], [46] or FO cumulants only [1]. Contrary to the methods based on HO cumulants only [1], the methods using SO cumulants [5], [6], [11], [18], [32], [34], [36], [46], [48] attempt to identify an orthogonalized version of the mixture. In practice, the latter converge faster as a function of the number of samples provided that the noise covariance is known, while the former are asymptotically insensitive to the presence of a Gaussian noise with unknown covariance.

Nevertheless, in practical fields, the overdetermined mixture assumption does not always hold. For example, in radiocommunications when the probability of receiving more sources than sensors increases with the reception bandwidth, then it can be necessary to solve problem 1 in the underdetermined case, for which \(P > N\). Such a context may also occur in biomedical engineering, especially when the number of electrodes would be strongly reduced in order to propose an ambulatory recording system [38]. Identifiability results on underdetermined source separation were introduced a few years ago [15], [45]. They showed that underdetermined mixtures could be identified up to a trivial matrix, that is a diagonal matrix and a permutation, provided that all sources are non-Gaussian. Many methods [2], [12], [16], [17], [24], [29], [30] have been developed since then. They use implicitly [12], [17], [29], [30] or explicitly [16], [24] a CANonical Decomposition (CAND) [31] of either SO [29], FO [12], [16], [17], [30] or Higher Order (HO) [2], [24] cumulants of the data. They decompose either one cumulant array [2], [16], [24], [30] or several ones at a given statistical order [12], [29]. In fact, CAND is an extension of the Singular Value Decomposition (SVD) to the case of multi-way arrays without requiring any orthogonality constraint. Other algorithms [14],

Manuscript received October 10, 2008; accepted August 16, 2009. First published September 09, 2009; current version published January 13, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alper Tunga Erdogan. This work is supported by the French Government, in one part by the ANR DECOTES Contract (06-BLAN-0074), in second part by the mv-EMD Contract (BLAN-07-0314-02) and in third part by the Brittany region Contract ACOMB-SOLARE. The authors are with the INSERM U642, Rennes F-35000, France, and the Université de Rennes 1, LTSI, Rennes F-35000, France.
take advantage of pseudo-cumulants, which are given by the (weighted) derivatives of the second characteristic function at some points different from the origin. However, all previous methods have some limits. Indeed, under some computational cost constraints and for \( N \geq 3 \) they do not succeed in exploiting all the information contained in the statistical tools they used, except the FOOBI2 method [30]. But the latter methods have some limits. Indeed, under some computational complexity. From a signal processing viewpoint, the proposed classical cumulant-based techniques. Secondly, from a numerical complexity. Thus, cumulants are presented in section II. In addition, the attractive properties of cumulants which make them easy to handle [23], [33]. In particular, they can be expressed as a function of moments of smaller order, which is the mathematical expectation, \( E[ ] \) of \( q-1 \) delays, \( \xi(\omega) \) equals \( \pm 1 \) with the conventions, \( \xi(0) = 1 \), \( \tau_0 = 0 \), \( x^1 = x \) and \( x^{-1} = x^* \) denoting by * the complex conjugate operator.

It is noteworthy that the estimation of cumulants using equation (2) is appropriate when the processed sources are stationary. But, in practice, sources are rarely so. In such a case, time averaged cumulants have to be used instead of (2), such as

\[
\mathcal{C}^{n_2, \ldots, n_q, \mathcal{O} \mathcal{V}}[\mathcal{T}] = \langle \text{Cum} \{ x_{n_1}[k], x_{n_2}[k-\tau_1]^{(1)}, \ldots, x_{n_q}[k-\tau_{q-1}]^{(q-1)} \rangle >
\]

where \( < . > \) is the mathematical expectation, \( x_n[k-\tau] \) is the \( n \)-th component of vector \( x[k-\tau] \), \( \tau \) is a set \( \{ \tau_1, \ldots, \tau_{q-1} \} \) of \( q-1 \) delays, \( \xi(\omega) \) equals \( \pm 1 \) with the conventions, \( \xi(0) = 1 \), \( \tau_0 = 0 \), \( x^1 = x \) and \( x^{-1} = x^* \) denoting by * the complex conjugate operator.

Definition 1: A partition of a set \( \Omega(q) = \{ 1, 2, \ldots, q \} \) of integers is a set of \( M \) (\( 1 \leq M \leq q \)) nonempty disjunctive subsets whose union is equal to \( \Omega(q) \).

Now, let \( \Omega^{(1)}_{\Omega}(M, q) \) be the \( m \)-th (\( 1 \leq m \leq M \)) subset of the \( j \)-th (\( 1 \leq j \leq J(M, q) \)) partition of size \( M \) of \( \Omega(q) \). This notation is illustrated in the following example:

Example 1: The different partitions of set \( \{ 1, 2, 3 \} \) are:

\[
\Omega^{(1)}_{\Omega}(1, 3) = \{ \{ 1, 2, 3 \} \}
\]

\[
\Omega^{(1)}_{\Omega}(2, 3), \Omega^{(2)}_{\Omega}(2, 3) = \{ \{ 1, 2 \}, \{ 3 \} \}
\]

\[
\Omega^{(1)}_{\Omega}(3, 3), \Omega^{(1)}_{\Omega}(3, 3), \Omega^{(3)}_{\Omega}(3, 3) = \{ \{ 1 \}, \{ 2 \}, \{ 3 \} \}
\]

with \( J(1, 3) = J(3, 3) = 1 \) and \( J(2, 3) = 3 \) different partitions.

Thus, we have the following proposition:

Proposition 1: Let \( \{ x[k] \}_{k \in \mathbb{N}} \) be an \( N \)-dimensional random process. Its \( q \)-th order delayed cumulants are given by:

\[
\text{Cum} \{ x_{n_1}[k], x_{n_2}[k-\tau_1]^{(1)}, \ldots, x_{n_q}[k-\tau_{q-1}]^{(q-1)} \} = \sum_{M=1}^{q} (-1)^{(M-1)}(M-1)! \prod_{j=1}^{M} \left[ \prod_{n_1=\omega} \prod_{n_q=\Omega^{(M, q)}(M, q)} x_{n_q}[k-\tau_{q-1}]^{(\omega-1)} \right] \]

where \( E[ ] \) is the mathematical expectation, \( x_n[k-\tau] \) is the \( n \)-th component of vector \( x[k-\tau] \), \( \tau \) is a set \( \{ \tau_1, \ldots, \tau_{q-1} \} \) of \( q-1 \) delays, \( \xi(\omega) \) equals \( \pm 1 \) with the conventions, \( \xi(0) = 1 \), \( \tau_0 = 0 \), \( x^1 = x \) and \( x^{-1} = x^* \) denoting by * the complex conjugate operator.

II. STATISTICAL TOOLS

Cumulants have many attractive properties which make them easy to handle [23], [33]. In particular, they can be expressed as a function of moments of smaller order, which is shown by the Leonov and Shiryayev formula [33], [43]. The latter formula is well-known in the case of non-delayed cumulants [33], [43]. But since in the following we will use the time dependence of each source through cumulants, we propose to rewrite this formula hereafter in the case of delayed cumulants. To this end, let’s give some notations and definitions.

Definition 1: A partition of a set \( \Omega(q) = \{ 1, 2, \ldots, q \} \) of integers is a set of \( M \) (\( 1 \leq M \leq q \)) nonempty disjunctive subsets whose union is equal to \( \Omega(q) \).

III. TOWARD A MULTI-WAY ARRAY FORMULATION

This section aims at showing how problem 1 can be solved using a joint CAND of several HO cumulant arrays through a multilinear algebra framework. Recall that multilinear algebra is defined as the algebra of q-way arrays (\( q > 2 \)), that is, the arrays whose elements are addressed by more than two indices. It is noteworthy that multilinear algebra can be related to tensor algebra through a multilinear map, called Segre map [13]. Now some definitions are necessary in order to understand how the BUMI problem can be reformulated in a multilinear algebra framework. First let’s begin by introducing a very useful product operator:

Definition 2: The outer product \( \mathbf{T} = \mathbf{u}^{(1)} \circ \cdots \circ \mathbf{u}^{(q)} \) of \( q \) (\( q \geq 2 \)) vectors \( \mathbf{u}^{(i)} \in \mathbb{C}^{N_i} \) (\( 1 \leq i \leq q \)) is a rank-1 array
where \( k(T) \) denotes the kruskal rank of \( T \) defined by the maximal number \( r \) such that any set of \( r \) columns of \( T \) is linearly independent [29]. Kruskal’s condition was firstly proposed for real 3-way arrays. Later it was extended to arrays with order higher than three [40] and for complex HO arrays [41]. The CAND uniqueness of specific 3-way arrays was also studied. For instance the case when one loading matrix has a full column rank [21] and the case where the HO array is tall in one direction [31]. In practice, the computation of CAND may require to handle unfolding matrices or vectors of multi-way arrays. A useful way to unfold square even order multi-way arrays is described below.

Definition 5: Let \( T \) be a square 2q-way \((q > 1)\) array of dimension \( N \), and let \( \ell \) be the integer part of \( q/2 \). Then the \((i,j)\)-th component of the unfolding matrix \( \text{mat}_1(T) \) of size \((N^q \times N^q)\) is given by:

\[
(\text{mat}_1(T))_{i,j} = T_{n_1, \ldots, n_q - \ell, n_q - \ell + 1, \ldots, n_q, n_q + 1, \ldots, n_2 - \ell, n_2 - \ell + 1, \ldots, n_2} \tag{6}
\]

where \( i = (n_1 - 1)N^q - 1 + \cdots + (n_q - \ell - 1)N^\ell + (n_2 - \ell - 1)N^\ell - 1 + \cdots + (n_2 - 1)N + n_2 \) and \( j = (n_q + 1 - 1)N^q - 1 + \cdots + (n_q - 1)N + n_q \).

For instance, let \( T \) be a square 6-way array of dimension \( N \). Then \( \ell = 1 \) and the \((i,j)\)-th component of the unfolding matrix \( \text{mat}_1(T) \) is given by:

\[
(\text{mat}_1(T))_{i,j} = T_{n_1, n_2, n_3, n_4, n_5, n_6} \tag{7}
\]

where \( i = (n_1 - 1)N^2 + (n_2 - 1)N + n_6 \) and \( j = (n_4 - 1)N^2 + (n_5 - 1)N + n_3 \).

Another way to transform higher order arrays into matrices is presented hereafter and is illustrated by figure 2.

Definition 6: Let \( T \in \mathbb{C}^{N_1 \times \cdots \times N_q} \) be a q-way \((q > 2)\) array. Then the \((n_1, n_q)\)-th component of the unfolding matrix \( \text{mat}_2(T) \) of size \((N^q \times N^q)\) associated to the \( i \)-th mode \((1 \leq i \leq q)\) of \( T \) is given by:

\[
(\text{mat}_2(T))_{n_1, n_q} = T_{n_1, \ldots, n_i - 1, n_i, n_i + 1, \ldots, n_q} \tag{8}
\]

where \( m = (n_i - 1)N_{i+1} \cdots N_q N_i - 1 + (n_i + 1 - 1)N_i - 1 + \cdots + (n_1 - 1)N_2 N_3 \cdots N_{i-1} + (n_2 - 1)N_3 N_4 \cdots N_{i-1} + \cdots + n_{i-1} \).

In addition, let’s introduce the following q-way array-to-vector transformation:

Definition 7: Let \( T \in \mathbb{C}^{N_1 \times \cdots \times N_q} \) be a q-way \((q > 1)\) array. Then the components of the \( N_1 \cdot \cdots \cdot N_q \)-dimensional vector \( \text{vec}(T) \) are defined by:

\[
\text{vec}(T)_{(n_1 - 1)N_2 \cdots N_q + (n_2 - 1)N_3 \cdots N_q + \cdots + n_q} = T_{n_1, \ldots, n_q} \tag{9}
\]

Conversely, let \( \text{unvec} \) be the inverse operator such as \( \text{unvec}(\text{vec}(T)) = T \).

Hermitian q-way arrays are considered in this paper, hence the following definition:
Definition 8: A complex q-way (q ≥ 2) array \( T \) is called hermitian when its CAND takes the following form:

\[
T = \sum_{p=1}^{P} \lambda_p u_p^{(1)} \circ u_p^{(2)} \circ \cdots \circ u_p^{(q)}
\]  

(10)

where the \( P \) values \( \lambda_p \) are real and where application \( \xi(i) \) is defined in proposition 1, that is, if the \( (P \times P) \) diagonal matrix \( \Lambda = \text{diag}([\lambda_1, \ldots, \lambda_P]) \) is real and if all the loading matrices of \( T \) are equal within a conjugate transformation.

Inserting the observation model (1) into (2) and using i) the statistical independence of the \( P \) source components, ii) the statistical independence between the sources and the noise and iii) the noise Gaussian distribution, we get for any integer \( q \geq 2 \):

\[
C_{2q}[\tau] = \sum_{p=1}^{P} C_{2q,p}[\tau] a_p^{\tau} \otimes a_p^{\tau}
\]  

(11)

where \( C_{2q,p}[\tau] = \text{Cum}\{s_p[k], s_p[k - \tau_1], \ldots, s_p[k - \tau_{q-1}], s_p[k - \tau_q]^*, s_p[k - 2 \tau_q - 1]^*, \ldots, s_p[k - \tau_{2q-1}]^* \} \) and where \( a^{\tau} \) is the outer product of \( q \) vectors \( a \). Formula (11) conveys the multilinearity property enjoyed by cumulant and moment arrays [23]. Then, provided that the \( P \) sources have at least \( M \) non-zero \( 2q \)-th order delayed marginal cumulants, problem 1 can be reformulated as following:

Problem 2: Given \( M \) 2q-th \((q \geq 2)\) order cumulant arrays \( C_{2q} \{\tau^{(m)}\} \) of \( \{x[k]\}_{k \in \mathbb{Z}} \) associated to a set \( \{\tau^{(m)}\} \) of \((2q-1)\)-dimensional time lags, find its joint CAND.

But, is it really possible to compute such a joint CAND, especially when \( P \) is greater than the maximum number of sources which can be processed by the classical cumulant-based techniques [1], [2], [5], [6], [11], [12], [14], [16]–[18], [24], [29], [30], [32], [34], [36], [46], [48]? Yes, it suffices to choose the cumulant order \((2q)\) greater or equal than four and six using the approaches proposed in sections IV-B and IV-A, respectively. Nevertheless, for the sake of simplicity and readability, we limited the presentation of our methods and their identifiability study to a fixed statistical order \(2q\) equal to six and identical for all the proposed algorithms. An extension to all other order would be then straightforward from sections IV and V.

IV. TOWARD A 3-WAY ARRAY DECOMPOSITION

We present in this section a new family of methods, named SIBI (Sixth order Blind mixture Identification), in order to solve problem 2 when \( q = 3 \). This family is essentially composed of two classes of techniques, namely the SAD (Semi-Algebraic Decomposition) and ALS (Alternate Least Square) SIBI approaches. Both classes can be described through two steps.

The first step consists in identifying the \( P \) rank-1 arrays \( A_p^{(3)} \) given by:

\[
\forall 1 \leq p \leq P, \quad A_p^{(3)} = a_p \circ a_p \circ a_p^*
\]  

(12)

from a CAND of a special 3-way array \( \mathcal{H}_a \) of size \((N^3 \times N^3)\) whose \((\ell_1, \ell_2, m)\)-th entry is given by:

\[
\mathcal{H}_{\ell_1, \ell_2, m,w} = C_{n_1, n_2,n_3}^{(3)}(\tau^{(m)}) = \text{mat}_1(C_{6,x}[\tau^{(3)}])
\]  

(13)

where \( \ell_1 = n_1 + (n_2-1)N + (n_3-1)N^2 \) and \( \ell_2 = n_3 + (n_5-1)N + (n_4-1)N^2 \). On the other hand, \( \mathcal{H}_a \) can also be built by stacking up the \( M \) hexacovariance matrices \( C_{6,x}[\tau^{(m)}] \) of size \((N^3 \times N^3)\) in the third direction. From (11) and (13), we get:

Proposition 2: Given the \( P \) \( M \)-dimensional vectors \( b_p = [C_{6,p}[\tau^{(1)}], \ldots, C_{6,p}[\tau^{(M)}]]^T \), \( \mathcal{H}_a \) has the following CAND:

\[
\mathcal{H}_a = \sum_{p=1}^{P} a_p^{(3)} \circ a_p^{(3)*} \circ b_p
\]  

(14)

where \( a_p^{(3)} = a_p \otimes a_p \otimes a_p^* \) with \( \otimes \) the Kronecker product operator and where \( A^{(3)} = [a_1^{(3)}, \ldots, a_p^{(3)}] \), \( A^{(3)*} \) and \( B = [b_1, \ldots, b_p] \) are the three loading matrices of \( \mathcal{H}_a \).

Once the \( P \) \( N^3 \)-dimensional vectors \( a_p^{(3)} \) have been identified from the CAND of \( \mathcal{H}_a \), each array \( A_p^{(3)} \) is found by computing \( A_p^{(3)} = \text{unvec}(a_p^{(3)}) \).

The second step consists in canonically decomposing each rank-1 array \( A_p^{(3)} \) in order to identify each column vectors \( a_p \) of \( A \). It is noteworthy that the way to implement this second step is the same for all SIBI methods and is based on the following proposition:

Proposition 3: Each \((N \times N^2)\) unfolding matrix \( A_p^{(3)} = \text{mat}_2^{(1)}(A_p^{(3)}) \) is the horizontal concatenation of \( N \) \((N \times N)\) matrices, which are of the form \( a_p a_p^\top \) with \( \gamma \in \mathbb{C} \).

In fact, \( \gamma \) is equal to the \((n,p)\)-th component, \( A_{n,p} \), of \( A \) when the \( n \)-th horizontal matrix block of \( A_p^{(3)} \) is concerned. Consequently, the eigenvector jointly associated with the largest eigenvalue of each of the \( N \) matrix blocks of \( A_p^{(3)} \) allows to estimate \( a_p \). Such an eigenvector can be computed using a joint diagonalization procedure [7], [8], [49], [50].
A. The SAD-SIBI approach

A semi-algebraic solution is proposed hereafter in order to find decomposition (14). It consists in handling the $M$ frontal slices $\mathcal{H}_a(:, :, m) = C_{6,a}[\tau^{(m)}]$ instead of $\mathcal{H}_a$ itself. Then according to (6) and (11) we can easily show that:

**Proposition 4:** Each hexacovariance matrix $C_{6,a}[\tau^{(m)}]$ has the following algebraic structure:

$$C_{6,a}[\tau^{(m)}] = A^{(3)} \zeta_{6,a}[\tau^{(m)}] A^{(3)\dagger}$$

where $\zeta_{6,a}[\tau^{(m)}] = \text{diag}([C_{6,a_1}[\tau^{(m)}], \ldots, C_{6,a_P}[\tau^{(m)}]])$.

Since $A^{(3)}$ is not generally orthogonal, equation (15) does not really correspond to the eigenvalue decomposition of $C_{6,a}[\tau^{(m)}]$, except when this latter has been prewhitened. Let’s see how to perform such a prewhitening. Under assumptions 1 and 2 given by:

**Assumption 1:** The $(N^3 \times P)$ matrix $A^{(3)}$ is full column rank.

and:

**Assumption 2:** The $(M \times P)$ matrix $B$ is full column rank.

respectively, the following proposition holds:

**Proposition 5:** There is at least one rank-$P$ positive semidefinite matrix, $\overline{C}_{6,a}$, which is a linear combination of the $M$ matrices $C_{6,a}[\tau^{(m)}]$.

Its proof is given in appendix A. In some practical contexts, for example when the $P$ non-delayed source marginal Sixth Order (SixO) cumulants have the same sign $\epsilon$, such as in radiocommunications, a simple linear combination is given by

$$\overline{C}_{6,a} = \epsilon C_{6,a} [0]$$

where $0 = \{0, 0, 0, 0, 0\}$. Otherwise, in the general case, we have to resort to numerical methods such that those presented in [47], [51] to compute $\overline{C}_{6,a}$. In addition, it is noteworthy that, matrix $\overline{C}_{6,a}$ may be non-hermitian and then should be replaced by $(\overline{C}_{6,a} + \overline{C}_{6,a}^\dagger)/2$ in order to ensure the existence of its eigen-value decomposition. For the sake of simplicity, we will assume in the sequel that matrix $\overline{C}_{6,a}$ is hermitian. Then proposition 5 ensures that $\overline{C}_{6,a}$ has a square root, which can be computed as following:

$$\overline{C}_{6,a}^{1/2} = E_u A_u^{1/2}$$

where $E_u$ is the $(N^3 \times P)$ unitary matrix whose columns are the eigenvectors associated with the $P$ largest eigenvalues of $\overline{C}_{6,a}$ sorted in the diagonal matrix $A_u$. Since two square roots of a positive semidefinite matrix are equal up to a unitary matrix, we have:

$$\overline{C}_{6,a}^{1/2} = A^{(3)} \zeta_{6,a}^{1/2} U_u^\dagger$$

where $\zeta_{6,a}$, $\zeta_{6,a}^{1/2}$, $U_u$ are the linear combination described in proposition 5 of the $M$ matrices $([\tau^{(m)}] + [\tau^{(m)}]^\dagger)/2$, a square root of the positive definite diagonal matrix $\zeta_{6,a}$ and a unitary matrix. Consequently, the inverse matrix, $\overline{C}_{6,a}^{-1/2}$, of $\overline{C}_{6,a}$ allows for a prewhitening of each matrix $C_{6,a}[\tau^{(m)}]$, say each matrix $\overline{C}_{6,a}[\tau^{(m)}] = \overline{C}_{6,a}^{-1/2} C_{6,a}[\tau^{(m)}] (\overline{C}_{6,a}^{-1/2})^\dagger$ is unitarily similar to a diagonal matrix indexed by $m$ as described by the following equation:

$$\tilde{C}_{6,a}[\tau^{(m)}] = U_u D[\tau^{(m)}] U_u^\dagger$$

where $D[\tau^{(m)}] = \zeta_{6,a}^{1/2} \zeta_{6,a}^{-1}$ is diagonal. Thus, a joint orthogonal diagonalization with the well-known semi-algebraic JAD algorithm [7] allows for an identification of $U_u$ up to a unitary trivial matrix. Nevertheless, $U_u$ is identifiable provided that, according to the results of [5], the $M (2q - 1)$-dimensional delays $\tau^{(m)}$ are chosen such that:

**Assumption 3:** For each couple $(p, q)$ of sources, there exists at least one set $\tau^{(m)}$ of delays for which:

$$\zeta_{6,p,q}[\tau^{(m)}] / \zeta_{6,p,q} \neq \zeta_{6,p,q}[\tau^{(m)}] / \zeta_{6,p,q}$$

It is noteworthy that assumption 3 is true under assumption 2. Indeed, assumption 2 implies that the $(M \times P)$ matrix $B \zeta_{6,a}$ is full column rank and so that all columns of $B \zeta_{6,a}$ are different. Hence equation (19). Thus, under assumption 2, the JAD algorithm diagonalizes the $M$ matrices $C_{6,a}[\tau^{(m)}]$ by minimizing the following criterion:

$$\Psi_1(U) = \sum_{m=1}^{M} \sum_{p \neq q=1}^{P} \| (U^\dagger \overline{C}_{6,a}[\tau^{(m)}] U)_{p,q} \|^2$$

with respect to (w.r.t.) to a unitary matrix $U$ expressed as the product of Givens rotations [19]. Once matrix $U_u$ has been identified, matrix $A^{(3)}$ can be computed up to a trivial matrix multiplying matrix $\overline{C}_{6,a}^{1/2}$ (17) by the estimate of $U_u$. This approach will be referred to as SAD-SIBI-JAD in the sequel.

B. The ALS-SIBI approaches

Another way to find decomposition (14) consists in iteratively and alternatively looking for the three loading matrices $A^{(3)}$, $A^{(3)}_s$, and $B$ of $\mathcal{H}_u$ in the least square sense. More precisely, from proposition 2 and definition 6, we have:

**Proposition 6:** Given the three unfolding matrices $H^{\tau_1}_x = \text{mat}_2(\mathcal{H}_x)$, $H^{\tau_2}_x = \text{mat}_2(\mathcal{H}_x)$, and $H^{\tau_3}_x = \text{mat}_3(\mathcal{H}_x)$ of $\mathcal{H}_x$, we get $H^{\tau_1}_x = A^{(3)} (A^{(3)}_s \odot B)$, $H^{\tau_2}_x = A^{(3)}_s (B \odot A^{(3)}_s)^\dagger$ and $H^{\tau_3}_x = B (A^{(3)} \odot A^{(3)}_s)^\dagger$ where $\odot$ is the Khatri-Rao product operator (columns-wise kronecker product) [42].

So, the following criterion can be minimized w.r.t. each of the three matrices $T^{(1)}, T^{(2)}$, and $T^{(3)}$ in order to compute the loading matrices $A^{(3)}$, $A^{(3)}_s$, and $B$ of $\mathcal{H}_u$ up to a trivial matrix:

$$\Psi_3(T^{(1)}, T^{(2)}, T^{(3)}) = \| H^{\tau_1}_x - (T^{(1)} (T^{(2)} \odot T^{(3)})^\dagger) \|^2$$

using the Alternate Least Square (ALS) procedure [42] where $\| \cdot \|$ denotes the Frobenius norm, giving rise to the ALS-SIBIUNS method. In fact, at each iteration $i$, we compute the three estimates $T^{(i)}$ ($1 \leq i \leq 3$) one at a time by fixing the others as described below:

$$T^{(1)} = H^{\tau_1}_x ((T^{(2)}_{it} \odot T^{(3)}_{it-1})^\dagger)$$

$$T^{(2)} = H^{\tau_2}_x ((T^{(1)}_{it-1} \odot T^{(3)}_{it})^\dagger)$$

$$T^{(3)} = H^{\tau_3}_x ((T^{(1)}_{it} \odot T^{(2)}_{it})^\dagger)$$

where $T^d$ denotes the pseudo-inverse of $T$. Consequently, matrices $T^{(1)}$, $T^{(2)}$, and $T^{(3)}$ should converge to matrices...
$A^{(3)}$, $A^{(3)*}$ and $B$, respectively, provided that the CAND
(14) is unique, which is ensured under assumptions 2 and the
following assumption:

Assumption 4: The $(N^6 \times P)$ matrix $A^{(1)} \otimes A^{(3)*}$ is full
column rank.

It is noteworthy that one is entitled to expect some con-
vergence cycles in which all factors will evolve in the same
direction [39] and consequently a slow convergence of the
ALS-SIBI_{UNS} method. An improved version of the ALS
scheme, named Line Search (LS) ALS, was proposed (see
[39] and the references therein) in order to exit from these
convergence cycles. This procedure is based on the relaxation
factor concept, which consists in extrapolating the estimated
loading matrices many iterations ahead. Then, we decided to
apply this LS procedure to the ALS-SIBI_{UNS} method, giving
rise to the LSALS-SIBI_{UNS} algorithm. The extrapolation of the
$i$-th estimated loading matrix of $\mathcal{H}_e$ is given by:

$$
T^{(i)}_{\text{new}} = T^{(i)}_{it-2} + \beta^{(i)}(T^{(i)}_{it-1} - T^{(i)}_{it-2})
$$

(23)

where $T^{(i)}_{\text{new}}$ is the matrix that will be used in the $i$-th
iteration instead of $T^{(i)}_{it-1}$, the term $(T^{(i)}_{it-1} - T^{(i)}_{it-2})$ defines
the direction of the cycle and $\beta^{(i)}$ is the relaxation factor
associated to the $i$-th loading matrix. In practice, the triplet
$(\beta^{(1)}, \beta^{(2)}, \beta^{(3)})$ of relaxation factors is chosen such that
$\beta^{(1)} = \beta^{(2)} = \beta^{(3)} = \frac{i}{1/n}$ where $n$ is a fixed integer
[39]. Recent works [39] showed that the convergence speed of the
LSALS procedure could be also increased, especially
when loading matrices are strongly correlated. This induced an
enhanced version of the LSALS algorithm, named Enhanced
Line Search (ELS)ALs, in the real [39] and the complex [35]
fields. More precisely, the ELS procedure consists, at each
iteration $it$ of the LSALS method, in finding the optimal
triplet $(\beta^{(1)}, \beta^{(2)}, \beta^{(3)})$. So, when the ELS scheme is applied to
LSALS-SIBI_{UNS} giving rise to the ELSALS-SIBI_{UNS} method,
the following criterion:

$$
\Psi_3(\beta) = \|H^{(1)}_x - G^{(1)}_{(it)}(G^{(2)}_{(it)} \otimes G^{(3)}_{(it)})^\dagger\|_F
$$

is minimized at each iteration $it$ w.r.t. $\beta^{(1)}$, $\beta^{(2)}$ and $\beta^{(3)}$
where $G^{(1)}_{(it)} = T^{(1)}_{it-2} + \beta^{(1)}(T^{(1)}_{it-1} - T^{(1)}_{it-2})$. The ELS scheme
allows to attain the final solution of a given cycle in only one
step. Note that the ELS procedure was implemented in the case
of three equal relaxation factors only, which is suboptimal but
less time-consuming.

However, by applying the previous ALS-like procedures to
$\mathcal{H}_e$, it amounts to assume a total independence between the
loading matrices, which is clearly wrong in our case. Thus
an alternative way of the ALS minimization would consists in
using the symmetries that exist between some loading
matrices of the considered multi-way array. In our case, the
first and the second loading matrices of $\mathcal{H}_e$ are equal up
to a complex conjugate as shown in (14). The ACDC-like
procedures [49], [39, Appendix 7.2] exploit such symmetries.
In fact the ACDC algorithm [49] minimizes the following
weighted least square criterion w.r.t. $V$ and the $M$ diagonal
$(P \times P)$ matrices $A[\tau^{(m)}]$:

$$
\Psi_2(V,A) = \sum_{m=1}^{M} w_m \left\| C_{6,m}[\tau^{(m)}] - VA[\tau^{(m)}]V^H \right\|_F^2
$$

(25)

where $w_m \in \mathbb{R}^{+}$. More particularly, the last minimization
problem is devised into two steps. The first one, named
Alternating Columns (AC), looks for the minimizer of (25)
with respect a selected column vector $v_p$ of $V$ keeping the $p-1$
other column vectors and the $M$ matrices $A[\tau^{(m)}]$ fixed. The second
step, called Diagonal Centers (DC), minimizes equation (25)
with respect the diagonal matrices $A[\tau^{(m)}]$ keeping $V$ fixed. So,
the ACDC algorithm alternates, until convergence, between the
$P$ consecutive AC sweeps, one by column vector of $V$, and the
DC step. Note that, an efficient way to initialize the ACDC
algorithm was recently proposed [50]. In the following, we
will refer to as the ALS-SIBI_{ACDC} method when the ACDC
procedure is used. But, are the ACDC-like procedures [49],
[39, Appendix 7.2] the only way to decompose $\mathcal{H}_e$ in a least
square sense using its symmetries? No, there is another way
such as the solution that we propose hereafter. Furthermore,
we combined this new 3-way array CAND method with the
ELS scheme in order to accelerate its convergence, giving rise
to the ELSALS-SYM procedure. In fact, the latter is mainly
based on the following proposition:

Proposition 7: Let $\mathcal{H}_e$ be the 3-way array of size $(N^3 \times
N^3 \times P)$ whose third $(P \times N^6)$ unfolding matrix is given by
$H^{(3)}_x = B^H H^{(3)}_x$. Then the $p$-th $(N^3 \times N^3)$ frontal slice of $\mathcal{H}_e$
$H^{(:,; \cdot ;p)}_e$ is a rank-1 matrix equal to $A^{(3)}_{(p)} \otimes \Phi^{(3)}_p$.

The proof is straightforward. Indeed, from proposition 2 we
get $H^{(3)}_x = (P(A^{(3)} \otimes A^{(3)*}))^T$, which implies that the $p$
frontal slice of $\mathcal{H}_e$ is equal to $A^{(3)} \text{diag}([I_p(p,:)]A^{(3)*})$
where $I_p(p,:)$ is the $p$-th row of the $(P \times P)$ identity matrix
$I_p$. Consequently, the eigenvector associated with the largest
eigenvalue of $H^{(:,; \cdot ;p)}_e$; $\Phi^{(3)}_p$ allows for an identification of the $p$
column vector of $A^{(3)}$ up to a scalar factor. Now, let’s see how
the ELSALS-SYM scheme uses this result. First the estimate,$T^{(3)}_{it-1}$, of the third loading matrix, $B$, of $\mathcal{H}_e$ is computed at
iteration $it - 1$ from equation (22). Next, using proposition 7,
we compute the $(P \times N^6)$ matrix $\tilde{T}^{(3)}_{it-1} = T^{(3)}_{it-1} H^{(3)}_x$
and build the 3-way array $\tilde{T}^{(3)}_{it-1}$ of size $(N^3 \times N^3 \times P)$
whose third $(P \times N^6)$ unfolding matrix is equal to $T^{(3)}_{it-1}$.
So, the eigenvector associated with the largest eigenvalue of the
$p$-th frontal slice of $\tilde{T}^{(3)}_{it-1}$ gives the $p$-th column vector of $T^{(3)}_{it-1}$. Thus, $T^{(2)}_{it-1}$ is computed as the conjugate of $T^{(1)}_{it-1}$. Eventually, the ELS scheme is used in order to accelerate the
convergence of the proposed algorithm. Note that all these
steps are repeated until convergence.

The convergence speed of the five proposed ALS-SIBI
approaches, namely ALS-SIBI_{UNS}, LSALS-SIBI_{UNS}, ELSALS-
SIBI_{UNS}, ALS-SIBI_{ACDC} and ELSALS-SIBI_{SYM}, is studied
now. More precisely, we display the median value of $\Psi_3$
(21) over the independent trials as a function of the number
of iterations at the output of the five ALS-SIBI methods. In
the next method, the latter methods are used to canonically decompose
the 3-way array $\mathcal{H}_e$ built from equation (14) with $M = 16$
5-dimensional delays \( \tau^{(m)} = [r_1^{(m)}, r_2^{(m)}, r_3^{(m)}, r_4^{(m)}, 0] \) and the estimated SixO delayed marginal cumulants of \( P = 3 \) angularly close QPSK sources. The latter sources are assumed to impinge on a Uniform Linear Array (ULA) of \( N = 2 \) sensors. In addition, all the considered approaches are initialized using the truncated HOSVD procedure [28]. Figure 3 shows, on the one hand, the good contribution of the ELS procedure to the ALS-SIBI method and on the other hand the interest for the ELSALS-SIBI algorithm in using symmetries of \( H_x \) in the way we propose.

V. IDENTIFIABILITY

An identifiability study of the SIBI family is proposed hereafter under assumptions 1-3. In particular, the \( M \) time lag vectors \( \tau^{(m)} \) are chosen such that a well-conditioned matrix \( B \) is obtained. Furthermore, we have to select at least an \( M = N^{2q} \) time lag vectors. Note that, in practice, the choice of these time lag vectors relies on some prior information on sources. Our identifiability study is fourfold. First from a BUMI viewpoint, it gives the maximum number, \( P_{max} \), of sources which can be processed by the SIBI family. Secondly, from a numerical analysis viewpoint, it tackles the identifiability of the SAD, ACDC and ELSALS\textsubscript{SYM} procedures when a 3-way array with special symmetries is considered. Next, from a multilinear algebra viewpoint it gives some insights on the uniqueness of a joint CAND of several hermitian multilinear arrays. Finally, from the VA theory viewpoint, we extend the VA concept [9] to the case of a combination of several VA’s. Recall that the VA theory relies on the existence of \( N_{2q} \) sensors which virtually appear when using the \( 2q \)-th order statistical matrix \( C_{2q,x} \) [9]. Regardless the application area, \( N_{2q} \) is equal to the maximal rank, \( r_{max} \), of the \( (N^q \times N) \) matrix \( A^{\otimes q} \):

\[
A_{\otimes q} = r_{max}(A^{\otimes q}) = r_{max}(A^{\otimes(q+1)} \odot A^{\otimes q}) \tag{26}
\]

where \( A^{\otimes q} \) denote the integer part of \( q/2 \) and the Khatri-Rao product of \( q \) matrices \( A \), respectively. Some values of \( N_{2q} \) in radiocommunications context are given in [9] for several kinds of antennas. For instance, when a ULA of \( N \) sensors is used, \( N_{2q} \) is given by \( N_{2q} = q(N-1) + 1 \).

Let’s now consider the SAD-SIBI\textsubscript{JAD} method. \( P_{max} \) is the maximum value of \( P \) for which assumptions 1 and 2 are true. Then we have:

\[
P_{max} = r_{max}(A^{(3)}) = N_{6} \tag{27}
\]

Regardin the five iterative ALS-SIBI approaches, their identifiability study consists in looking for the rank maximal of the 3-way array \( H_x^{(14)} \) they decompose. This should be achieved provided that the CAND of \( H_x \) is unique. Finding this rank is usually a matter of trial and error [31]. Nevertheless, under assumption 2 and according to the results of [21] we have:

\[
P_{max} = r_{max}(H_x^{(3)}) = r_{max}(B(A^{(3)} \odot A^{(3)})) \tag{28}
\]

where \((M \times N^q) H_x^{(3)}\) denotes the unfolding matrix associated to the third direction of \( H_x^{(14)} \). For \( M \geq N^q \), the maximum rank of \( H_x^{(3)} \) is generically equal to \( N^q \). Recall that a property is well-called generic when it holds everywhere except for a set of Lebesgue measure 0. However, in some applications such as in radiocommunication contexts, matrix \( H_x^{(3)} \) may belong to this set. In this case, the previous property does not hold generally. Intrinsically, we have:

\[
r_{max}(H_x^{(3)}) \leq r_{max}(A^{(6)}) = N_{12} \tag{29}
\]

where \( A^{(6)} = A^{(3)} \odot A^{(3)\ast} \). But, since according to [9] \( N_{12} \) is strictly lower than \( N^q \), the number maximal of sources which can be processed using the ALS-SIBI approaches is then given by:

\[
P_{max} = r_{max}(A^{(6)}) = N_{12} \tag{30}
\]

So from a BUMI viewpoint and according to equations (27) and (30) the SIBI methods can process more than \( N_q \) sources. Note that \( N_q \) is the maximum number of sources the classical BUMI cumulant-based methods [2], [12], [16], [17], [24], [29], [30] can process. Table I gives the number \( P_{max} \) as a function of \( N \) for a ULA and for several cumulant-based methods. From a numerical analysis viewpoint, we show that: i) the SAD scheme is limited to 3-way arrays whose rank is lower than their smallest dimension. ii) Both the ACDC and ELSALS\textsubscript{SYM} procedures succeed generically in performing the CAND of a rank-\( N^q \) \((N^q \times N^q \times M)\) array if \( M \geq N^q \). Consequently, a more relaxed uniqueness condition can be obtained when a joint CAND of several hermitian multiway arrays is performed. Since according to [9] \( N_0 = r_{max}(C_{x} \otimes \{0\}) \), equation (30) shows that a maximal rank of \( N_{12} \) can be achieved when \( C_{x} \otimes \{0\} \) and at least \( N_{12} - 1 \) other arrays \( C_{x} \otimes \{\tau\} \) with the same loading matrices are simultaneously considered. This result extends the virtual array theory [9] to the case where several 2q-th order cumulant array are simultaneously used. Hence, a twofold increase of the virtual array aperture.
VI. NUMERICAL COMPLEXITY

The numerical complexity of the SIBI family is addressed hereafter. It is computed and compared to the one of classical BUMI methods, namely SOBI [5], JADE [6], FastICA [20, chapter 8], COM2 [11], FOBIUM [30], FOBI2 [30], FOBIUM [17] and 6-BIOME [2]. More particularly, it is computed in terms of number of floating point (complex) operations (flops) required to identify the \((N \times P)\) mixing matrix \(A\) from \(K\) data snapshots. Note that a complex flop is defined as the sum of a complex multiplication and a complex addition. In practice, only multiplications are counted which does not affect the magnitude of the numerical complexity. Table II shows the computational complexity of the aforementioned methods where \(f_s(N) = O[N^4/8]\) and \(f_a(N) = O[N^6/2]\) denote the number of free entries of \(C_{1,x}\) and \(C_{6,x}\), respectively. In addition, \(D\) and \(I\) denote the desired resolution in the unit disk for the COM2 approach and the number of sweeps used in the joint diagonalization algorithms, respectively. \(J\), \(J_1\), \(J_2\) and \(J_3\) are the maximal numbers of iterations used by FastICA, ELSALS-SIBI\textsubscript{uns}, ELSALS-SIBI\textsubscript{sym} and ALS-SIBI\textsubscript{ACDC}, respectively. \(J_4\) is the maximal number of iterations used for the estimation of the relaxation factor in ELSALS-SIBI\textsubscript{uns} and ELSALS-SIBI\textsubscript{sym}. Eventually, \(M\) represents the number of delay lags used in SOBI, FOBIUM and the SIBI techniques, \(M_1 = \max\{N^6, P\}, m_1 = \min\{N^6, P\}, M_2 = \max\{M, P\}\) and \(m_2 = \min\{M, P\}\).

Table II shows the minimum numerical complexity of the SIBI family and the aforementioned classical BUMI methods, as a function of the number of sources, for a ULA and 1000 data snapshots.

![Fig. 4. Minimal complexity of the SIBI family and other BUMI methods as a function of the number of sources, for a ULA and 1000 data snapshots.](image)

VII. SIMULATION RESULTS

Five performance evaluation studies of the SIBI family are presented hereafter, and more precisely the performance of SAD-SIBI\textsubscript{JAD}, ALS-SIBI\textsubscript{ACDC} and ELSALS-SIBI\textsubscript{SYM} for both underdetermined and overdetermined mixtures of sources. The performance analysis was achieved using the criterion \(D(A, \tilde{A}) = (\alpha_1, \alpha_2, \ldots, \alpha_P)\), which measures a pseudo-distance between mixture \(A\) and its estimate \(\tilde{A}\) where \(\alpha_p = \min_{1 \leq i \leq P} d(a_p, \tilde{a}_i)\) with \(d\) the pseudo-distance between vectors defined by [17]:

\[
d(u, v) = 1 - \|u^*v\|^2 / \|u\|^2 \|v\|^2
\]

It is noteworthy that only criterion \(\alpha_1\) is displayed in figures 5, 7, 8 and 9 since the \(P - 1\) other criteria \(\alpha_p\) have shown identical results. The \(P\) sources considered in this section are Quadrature Phase Shift Keying (QPSK) linearly modulated with pulse shape filter corresponding to a 1/2-Nyquist filter with a roll-off of 0.3. The latters are assumed to impinge on a ULA of \(N\) sensors and are chosen to be well angularly separated.

Both first studies consist in evaluating the performance of the SIBI family in a strongly underdetermined mixture which
cannot be processed by any classical cumulant method based on at most order 6, say \( P \geq 4 \) sources from a ULA of only \( N = 2 \) sensors. The sources have the same Signal-to-Noise Ratio (SNR) equal to 15 dB. On the one hand, figure 5 shows the variation of \( \alpha_1 \) at the output of the three SIBI methods as a function of the number of samples for \( P = 4 \) sources with carrier residuals such that \( f_c, T_e = 0 \), \( f_c, T_e = 0.25 \), \( f_c, T_e = 0.5 \). In addition, the performance was compared for two numbers \( M \) of sets \( \tau^{(m)} \) of delays, say \( M = 9 \) and \( M = 27 \). Figure 5 shows a good behavior of the SIBI family. Besides, the semi-algebraic SAD-SIBI\textsubscript{JAD} method seems to be insensitive to the increase in the number of used sets of delays, contrary to the iterative ALS-SIBI\textsubscript{ACDC} and ELSALS-SIBI\textsubscript{SYM} algorithms. As shown in section V, more statistical information may ensure the uniqueness of CAND and then a good quality of factor’s identification. Since better results are obtained for \( M = 27 \) sets of delays, the latter were used in the following studies. On the other hand, the case of \( P = 7 \) sources with residual carriers given by \( f_c, T_e = 0 \), \( f_c, T_e = 0.15 \), \( f_c, T_e = 0.3 \), \( f_c, T_e = 0.45 \), \( f_c, T_e = 0.6 \), \( f_c, T_e = 0.75 \), and \( f_c, T_e = 0.9 \) is considered in figure 6. Note that, in such a practical context, only ALS-SIBI\textsubscript{ACDC} and ELSALS-SIBI\textsubscript{SYM} are able to process as many sources, as mentioned in table I. More precisely, figure 6 shows the maximum \( \alpha_p \) named \( \alpha_{\text{max}} \), for each data sample at the output of ALS-SIBI\textsubscript{ACDC} and ELSALS-SIBI\textsubscript{SYM}, say the worst performance over all sources. 

<table>
<thead>
<tr>
<th>( N ): number of sensors, ( P ): number of sources, ( M ): number of time lags, ( L ): number of data samples, ( I ): number of sweeps, ( J, J_1, J_2, J_3, J_4 ): number of iterations in FastICA, in the ELSALS-SIBI\textsubscript{UNS}, ELSALS-SIBI\textsubscript{SYM} in the estimation of the relaxation factor in ELSALS-SIBI and in the ALS-SIBI\textsubscript{ACDC}, respectively</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COM2</strong></td>
</tr>
<tr>
<td>( \min {LN^2/2 + 4N^2/3 + PNL, 2LN^2 } + \min {12fJ_1(P)P^2 + 2IP^2 + 3LJ_4(P) + LP^2, 13ILP^2/2 } + 7IP^2/D^2 )</td>
</tr>
<tr>
<td><strong>FastICA</strong></td>
</tr>
<tr>
<td>( \min {LN^2/2 + 4N^2/3 + PNL, 2LN^2 } + J[2(P - 1)(P + L) + 5L(P + 1)/2] )</td>
</tr>
<tr>
<td><strong>SOBI</strong></td>
</tr>
<tr>
<td>( MLN^2/2 + 4LN^2/3 + P(N^2 + N^3) - N + 17P^2 + 4P^3) + MP(N - 1) + (17M + 75 + 4P + 4PM) + 2L^2 )</td>
</tr>
<tr>
<td><strong>JADE</strong></td>
</tr>
<tr>
<td>( \min {LN^2/2 + 4LN^2/3 + PNL, 2LN^2 } + \min {4P^3, 8P^4 + 3 } + 3LJ_4(P) + IP^2(75 + 21P + 4P^3) + 2L^2 )</td>
</tr>
<tr>
<td><strong>FOBIUM</strong></td>
</tr>
<tr>
<td>( 3MJ_1(N) + 2LP^3 + 3(N - P)^{4} + (N - 1)^2 + IP^2 + (P - 1)[4P^2 + 11P + 75] + (N - 1)/3 )</td>
</tr>
<tr>
<td><strong>FOOBI1</strong></td>
</tr>
<tr>
<td>( 3LJ_4(N) + 2LN^3/3 + (N - P)^2 + 3/2 + N^2P + 2N^2P^2 + P^3 + (P - 1)[4P^2 + 11P + 75]/2 + N^2P^2 + 2LP^2 + 21P + 4P^3) + MP(N - 1) )</td>
</tr>
<tr>
<td><strong>FOOBI2</strong></td>
</tr>
<tr>
<td>( 3LJ_4(N) + 2LN^3/3 + (N - P)^2 + 3/2 + N^2P + 2N^2P^2 + P^3 + (P - 1)[4P^2 + 11P + 75]/2 + N^2P^2 + 2LP^2 + 3MP(N - 1) )</td>
</tr>
<tr>
<td><strong>6-BIOME</strong></td>
</tr>
<tr>
<td>( 5LJ_4(N) + 2LN^3/3 + (N - P)^2 + 3/2 + N^2P + 2N^2P^2 + P^3 + (P - 1)[4P^2 + 11P + 75]/2 + N^2P^2 + 2LP^2 + 3MP(N - 1) )</td>
</tr>
<tr>
<td><strong>SAD-SIBI\textsubscript{JAD}</strong></td>
</tr>
<tr>
<td>( 5LMJ_6(N) + 4LN^3 + 3P + PMN^2 + 3P + PMN^2 + MP(N - 1) )</td>
</tr>
<tr>
<td><strong>ALS-SIBI\textsubscript{ACDC}</strong></td>
</tr>
<tr>
<td>( 5LMJ_6(N) + (2PN + 2N^3 + N + 4P^2 + P + (M + 1)PN^2) + 17P^2 + PIN(N - 1) )</td>
</tr>
<tr>
<td><strong>ELSALS-SIBI\textsubscript{UNS}</strong></td>
</tr>
<tr>
<td>( 5LMJ_6(N) = (P + 8P^2)(2P^2N + 3PMN^2 + 3PMN^2) + P^2 + 8PMN^2 + 2PMN^2 + 8PMN^2 + 6N^2M + 6N^2MP + 5P(N - 1) )</td>
</tr>
<tr>
<td><strong>ELSALS-SIBI\textsubscript{SYM}</strong></td>
</tr>
<tr>
<td>( 5LMJ_6(N) + (2PN + 2N^3 + N + 4P^2 + P + (M + 1)PN^2) + 17P^2 + PIN(N - 1) )</td>
</tr>
</tbody>
</table>

**TABLE II**

Numerical complexity of the SIBI family and classical cumulant-based methods in terms of complex flops. 

Fig. 5. \( \alpha_1 \) for a ULA of 2 sensors, 4 QPSK’s with the same SNR=15 dB and two numbers \( M \) of delays: \( M = 9 \) (black line) and \( M = 27 \) (grey line). 

Fig. 6. Variation of the worst \( \alpha_p \) criterion for each data sample for a ULA of 2 sensors and 7 QPSK’s of SNR= 15 dB.
of both considered methods. It appears that both methods are effectively able to process up to seven sources from a ULA of two sensors. In addition, in comparison with figure 5, ELSALS-SIBI$^{SYM}$ seems to be less sensitive to a strong increase of the number $P$ of sources even it also needs more samples to give a constant performance as $P$ increases.

The third study consists in comparing, in a more sympathetic underdetermined mixture case, the SIBI family and cumulant-based methods such as FOBIUM [17], 6-BIOME [2], as a function of the number of samples. For this purpose $N = 2$ sensors and $P = 3$ sources are considered with residual carriers given by $f_c, T_e = 0, f_c, T_e = 0.35, f_c, T_e = 0.7$ and an SNR of 15 dB. While FOBIUM and 6-BIOME appear to be efficient in figure 7, especially when the number of samples increases, the SIBI family shows a higher performance even for a small number of samples.

Now we evaluate the influence of SNR on the performance of the SAD-SIBI$^{JAD}$, ALS-SIBI$^{ACDC}$, ELSALS-SIBI$^{SYM}$, FOBIUM and 6-BIOME using the same application context as the one used in figure 7. Under the assumptions, figure 8 shows that the five methods have a similar behavior for small values of SNR. Nevertheless, the SIBI techniques give better results as soon as SNR increases beyond 0 dB.

Eventually, figure 9 shows the variations of $\alpha_1$ as a function of the spatial correlation factor of a Gaussian noise at the output of the three considered SIBI approaches and seven classical cumulant-based methods such as COM2 [11], FastICA [20, chapter 8], FOOBi1, SOBI [5], JADE [6], 6-BIOME and FOBIUM. For this purpose, we considered 1000 samples, $N = 3$ sensors and $P = 2$ sources with residual carriers such that $f_c, T_e = 0, f_c, T_e = 0.65$ and a SNR of 5 dB for all sources. Note that the Gaussian noise model used in this simulation is the sum of an internal noise $\nu_{in}[k]$ and an external noise $\nu_{out}[k]$ of covariance matrices $C_{\nu_{in}}^{in}[k, 0]$ and $C_{\nu_{out}}^{out}[k, 0]$, respectively, such that:

$$
C_{2, \nu}^{in}[0, r, q] = \frac{\sigma^2}{2} \delta(r - q) \quad C_{2, \nu}^{out}[0, r, q] = \frac{\sigma^2}{2} \rho^{(r-q)}
$$

where $\sigma^2$, $\rho$, $C_{2, \nu}[0, r, q] = C_{2, \nu}^{in}[0, r, q] + C_{2, \nu}^{out}[0, r, q]$ are the total noise variance per sensor, the noise spatial correlation factor and the $(r, q)$-th component of the total noise covariance matrix, respectively. The computer results show on the one hand how the performance of the SIBI family is stable, even for a strong noise spatial correlation. In one hand, the classical methods seem to be less robust, especially COM2, FastICA, JADE and SOBI which require a prior spatial whitening based on SO cumulants.

VIII. CONCLUSION

We proposed a new class of BUMI methods based on the joint CAND of a set of HO cumulant arrays. This class of algorithms is described through SixO cumulants, giving rise to two SixO subfamilies of techniques, namely the SAD-SIBI and the ALS-SIBI methods. More particularly, these two families perform a CAND of a 3-way array built from a set of SixO cumulant arrays but in a different way. Indeed, the former approach is algebraic while the latter is iterative. All approaches were theoretically studied especially in terms of identifiability and computational complexity. In particular, the identifiability analysis showed that the SIBI algorithms can process more sources than the classical cumulant-based methods. In addition, computer results showed the good behavior of the SIBI family against classical methods, for both overdetermined and underdetermined mixtures of sources. Besides, from a numerical analysis viewpoint, we deal with the identifiability and the convergence speed of several 3-way array decomposition procedures such as the symmetric ACDC scheme. We also show how to accelerate the iterative CAND of a 3-way array, which has special symmetries. Next, from a multi-linear algebra viewpoint we give some insights on the
for a ULA of several VAs with the same number of virtual sensors amounts to the case of a combination of several VAs. Indeed, the use of this allows us to extend the concept of Virtual Array (VA) to arrays compared to the CAND of only one array. In addition, Fig. 9.

Appendix A

Proof of Proposition 5

The full column rank of $B$ implies that there is at least one $M$-dimensional vector $\beta$ with values in the complex field such that:

$$ B'\beta = [1, 1, \ldots, 1]' \in \mathbb{C}^P $$

(33)

According to the definition of matrix $B$ given in proposition 2, (33) is equivalent to the following equation:

$$ \sum_{m=1}^{M} \beta_m \xi_{6\times}[\tau^{(m)}] = I_P $$

(34)

where $\beta_m$ is the $m$-th component of the $M$-dimensional column vector $\beta$ and $I_P$ is the $(P \times P)$ identity matrix. Then, under assumption 1, the following matrix:

$$ \sum_{m=1}^{M} \beta_m C_{6\times}[\tau^{(m)}] = A^{(3)} \left( \sum_{m=1}^{M} \beta_m \xi_{6\times}[\tau^{(m)}] \right) A^{(3)\dagger} $$

(35)

is a rank-$P$ positive semidefinite matrix equal to $A^{(3)} A^{(3)\dagger}$. Hence the result.

References


