Batch and Adaptive PARAFAC-Based Blind Separation of Convolutive Speech Mixtures

Dimitri Nion, *Member, IEEE*, Kleanthis N. Mokios, *Student Member, IEEE*
Nicholas D. Sidiropoulos, *Fellow, IEEE* and Alexandros Potamianos, *Member, IEEE*

***Abstract*— We present a frequency-domain technique based on PARAllel FACtor (PARAFAC) analysis that performs multichannel Blind Source Separation (BSS) of convolutive speech mixtures. PARAFAC algorithms are combined with a dimensionality reduction step to significantly reduce computational complexity. The identifiability potential of PARAFAC is exploited to derive a BSS algorithm for the under-determined case (more speakers than microphones), combining PARAFAC analysis with time-varying Capon beamforming. Finally, a low-complexity adaptive version of the BSS algorithm is proposed that can track changes in the mixing environment. Extensive experiments with realistic and measured data corroborate our claims, including the under-determined case. Signal-to-Interference ratio improvements of up to 6 dB are shown compared to state-of-the-art BSS algorithms, at an order of magnitude lower computational complexity.**

I. Introduction

Blind Source Separation (BSS) aims to estimate multiple source signals mixed through an unknown channel, using only the observed signals captured by a set of sensors. There are diverse potential applications of BSS in various areas, including speech processing, telecommunications, biomedical signal processing, analysis of astronomical data or satellite images, etc. In this paper, we focus on BSS of speech signals recorded in a reverberant environment. In this situation, multiple attenuated and delayed versions of each speaker signal are captured by each microphone, which results in a problem of blind separation of convolutive speech mixtures. This is a key problem in applications such as teleconferencing or mobile telephony, where multiple speaker separation or speaker-background separation can be crucial for human intelligibility and automatic speech recognition.

BSS techniques usually assume certain properties on the sources or the mixing system and capitalize on a separation criterion that imposes the same properties on their estimates. In BSS of speech signals, a significant attribute that can be exploited is the inherent non-stationarity of such signals. Speech signals are in fact considered to be non-stationary for durations greater than 40 ms [1]. Several BSS algorithms that exploit non-stationarity have been proposed in the simple case of instantaneous linear mixtures, e.g., [2]. In the more realistic case of convolutive linear mixtures, time-domain [3, 4] and frequency-domain [5–9] methods have been proposed. We refer to [10] for a categorization of existing convolutive BSS methods (see also Section II).

Exploiting the non-stationary nature of speech signals, the BSS problem can be solved via the use of Second-Order-Statistics (SOS), assuming uncorrelated sources. Thus, the problem reduces to estimation of the mixing matrix that minimizes a measure of total cross-correlation. If the mixing system is stationary, the solution can be obtained by considering multiple cross-correlation lags, which yields a Joint-Approximate-Diagonalization (JAD) problem [11, 12]. Such an approach was proposed in, e.g., [13], for BSS of instantaneous mixtures, and in, e.g., [5, 6, 8], for BSS of convolutive mixtures in the frequency domain. The main challenges towards engineering pragmatic BSS algorithms for convolutive speech mixtures in the frequency domain are the following:

i) Building a fast and robust separation algorithm that solves the JAD problem for each frequency bin.

ii) Dealing with under-determined cases, i.e., when the number of sources exceeds the number of microphones. This entails identifiability issues and requires appropriate cross-talk reduction techniques, which have not been properly addressed to date in this context.

iii) Effectively dealing with the frequency-dependent permutation and scaling ambiguity problems.

iv) Dealing with non-stationary mixing environments, i.e., solving the BSS problem adaptively.

In this paper, we propose original contributions for each of these four challenges. First, we show that solving a JAD problem for each frequency is equivalent to fitting a conjugate symmetric *parallel factor* (PARAFAC) model for each frequency. PARAFAC is a powerful multilinear algebra tool for tensor decomposition in a sum of rank-1 tensors. In this sense, PARAFAC is one possible generalization of the matrix SVD to higher-order tensors. PARAFAC was introduced in [14] in 1970 and slowly found its way in various disciplines such as Chemometrics and food technology [15], exploratory data analysis [16], wireless communications and array processing [17, 18], and BSS [19, 20]. In the context of this paper, exploitation of the algebraic structure of the PARAFAC model for each frequency allows a dimensionality-reduction step before the separation stage. This results in a far lower complexity than state-of-art JAD techniques [5, 6, 8], with guaranteed convergence.

Next, we show that, unlike state-of-art JAD algorithms, the strong uniqueness properties of PARAFAC allow us to identify the mixing matrix transfer function in certain under-determined cases. For the simpler case of instantaneous mix-

The authors are with the Department of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania, Greece. E-mails: {nion, kleanthis, nikos, potam}@telecom.tuc.gr. D. Nion was supported by a post-doctoral grant from the Délégation Générale pour l'Armement (DGA) via ETIS Lab., UMR 8051 (ENSEA, CNRS, Univ. Cergy-Pontoise), France. N. Sidiropoulos simply enjoyed applying PARAFAC to an engineering problem.

tures, an analogous result was established in [20]. We propose to build the de-mixing matrix by employing a time-varying Capon beamforming-based cross-talk reduction technique, and demonstrate good performance for under-determined cases.

The third contribution of this paper is a low-complexity technique to deal with the frequency-dependent permutation problem. Our method consists of clustering the (properly scaled) estimated source profiles via the *k-means* algorithm, after which the permutation matrices are estimated in a single step, in a non-iterative way. This clustering strategy results in a significant reduction of the complexity, compared to the fully iterative techniques proposed in [8, 21, 22], without sacrificing performance.

Finally, we derive an adaptive version of our batch blind speech separation algorithm, based on one of the adaptive algorithms that we have developed in [23] to track a PARAFAC decomposition. This is important to track changes in the acoustic environment (e.g., due to speaker movement), and it also yields complexity savings as a side benefit - thus bringing the overall solution closer to practice.

Preliminary results have appeared in conference form in [24, 25]. This journal version incorporates i) a much faster separation algorithm, ii) a novel permutation-matching algorithm, iii) a technique to deal with the under-determined case, iv) an adaptive version of the algorithm, and v) extensive experiments.

This paper is organized as follows. In Section II, we give the general formulation of the frequency-domain BSS problem in terms of JAD of a set of matrices for each frequency bin. In Section III, we establish the link between the JAD formulation and its equivalent PARAFAC reformulation and we report existing results concerning uniqueness of PARAFAC. In Section IV, we explain our approach for batch computation of the PARAFAC decomposition for each frequency bin. In Section V, we explain how scaling and permutation ambiguities can be corrected. In Section VI, we address the under-determined case and we show how a time-varying Capon beamforming technique can be employed for cross-talk reduction. In Section VII, we discuss an adaptive version of our batch algorithm. Section VIII reports numerical results, and Section IX summarizes our conclusions.

Notation. A third-order tensor of size $I \times J \times K$ is denoted by a calligraphic letter $\mathcal{Y}$, and its elements are denoted by $y_{i,j,k}$, $i = 1, \ldots, I$, $j = 1, \ldots, J$ and $k = 1, \ldots, K$. A bold-face capital letter $\mathbf{Y}$ denotes a matrix and a bold-face lower-case letter $\mathbf{y}$ a vector. The transpose, complex conjugate, complex conjugate transpose and pseudo-inverse are denoted by $\mathbf{Y}^T$, $\mathbf{Y}^*$, $\mathbf{Y}^H$ and $\mathbf{Y}^\dagger$, respectively. $\|\mathbf{Y}\|_F$ denotes the Frobenius norm of $\mathbf{Y}$. The Kronecker product is denoted by $\otimes$. The Khatri-Rao product (or column-wise Kronecker product) is denoted by $\odot$, i.e., $[\mathbf{a}_1, \ldots, \mathbf{a}_I] \odot [\mathbf{b}_1, \ldots, \mathbf{b}_I] = [\mathbf{a}_1 \otimes \mathbf{b}_1, \ldots, \mathbf{a}_I \otimes \mathbf{b}_I]$. The $P \times P$ identity matrix is denoted by $\mathbf{I}_P$. $E[\cdot]$ denotes the expectation operator. We will also use a Matlab-type notation for matrix sub-blocks, i.e., $[\mathbf{A}]_{l:m,n:p}$ represents the matrix built after selection of $m - l + 1$ rows of $\mathbf{A}$, from the lth to the mth, and $p - n + 1$ columns of $\mathbf{A}$, from the nth to the pth. $[\mathbf{A}]_{:,n:p}$ is used to denote selection of all rows and $[\mathbf{A}]_{l:m,:}$ to denote selection of all columns. Similarly, $\mathbf{y}(l : m)$ represents a selection of $m-l+1$ samples of the vector $\mathbf{y}$, from the lth to the mth.

II. Problem Statement

A. Data Model

Let us consider I mutually uncorrelated speaker signals $\mathbf{s}(t) = [s_1(t), \ldots, s_I(t)]^T$ captured by J microphones and denote by $\mathbf{x}(t) = [x_1(t), \ldots, x_J(t)]^T$ the recorded mixtures. The noise-free convolutive model is written as follows

$$\mathbf{x}(t) = \mathbf{H} \star \mathbf{s}(t) = \sum_{l=0}^{L-1} \mathbf{H}(l)\mathbf{s}(t-l), \tag{1}$$

where $\star$ is the linear convolution operator. The $J \times I$ matrix $\mathbf{H}(l)$ represents the mixing system at time-lag l. Its elements $h_{j,i}(l)$ are coefficients of the Room Impulse Response (RIR) between source i and microphone j, modeled as a Finite Impulse Response (FIR) filter. L denotes the maximum (unknown) channel length. To estimate the sources $\hat{\mathbf{s}}(t) = [\hat{s}_1(t), \ldots, \hat{s}_I(t)]^T$, the objective is to find an $I \times J$ approximate inverse-channel matrix $\mathbf{W}$, such that

$$\hat{\mathbf{s}}(t) = \mathbf{W} \star \mathbf{x}(t) = \sum_{k=0}^{K-1} \mathbf{W}(k)\mathbf{x}(t-k), \tag{2}$$

where K is the length of the inverse-channel impulse response.

To solve this problem, one can resort to a time-domain approach or a frequency-domain approach. In time-domain approaches, K should be chosen at least equal to the unknown true channel order L for all reflections to be modeled, and much larger than L for accurate estimation. Time-domain methods are sensitive to channel-order mismatch [10], and their identifiability properties are not adequately understood, especially in under-determined cases.

Frequency-domain BSS methods begin by mapping the problem to the frequency domain by applying the Discrete-Fourier-Transform (DFT) on the observed signals:

$$\mathbf{x}(t) = \mathbf{H} \star \mathbf{s}(t) \leftrightarrow \mathbf{x}(f, q) \approx \mathbf{H}(f)\mathbf{s}(f, q), \tag{3}$$

where f is a frequency index, $f = 1, \ldots, F$, q is a frame index, $\mathbf{x}(f, q) = [x_1(f, q), \ldots, x_J(f, q)]^T$ and $\mathbf{s}(f, q) = [s_1(f, q), \ldots, s_I(f, q)]^T$. The ith column of $\mathbf{H}(f)$ represents the spatial signature of the ith speaker in the frequency domain, at frequency f. Note that the approximation (3) is exact only for periodic signals $\mathbf{s}(t)$, or equivalently, if the time-convolution is circular. This approximation is satisfactory if F is significantly larger than the maximum length L of the mixing channels [6]. To limit the circularity effect, a spectral smoothing approach is commonly used [26]. In practice, we will compute the DFT of consecutive overlapping windowed frames (a Hanning window will be used).

The main advantage of a frequency-domain approach is to transform the initial convolutive time-domain model into a set of instantaneous BSS problems, for which several efficient algorithms have been proposed in the literature. However, the main difficulty with BSS in the frequency-domain is the need to cope with the permutation and scaling ambiguities, i.e., the mixing matrix is estimated up to an arbitrary permutation and scaling of its columns for each frequency. Before converting the estimated source signals back to the time domain, the scaling ambiguity must be compensated and a permutation matching procedure must be applied to associate the spectral components belonging to the same source. Different methods have been proposed to resolve the permutation ambiguity; see [10] for a recent survey. In Section V, we will propose a new variation of the the permutation correction techniques proposed earlier in [8, 21, 22]. This yields significant complexity reduction relative to the fully iterative methods in [8, 21, 22], without sacrificing performance.

Before proceeding further, we list our main assumptions:

Assumption 2.1- The speaker signals $\mathbf{s}(t)$ are zero-mean, mutually uncorrelated.

Assumption 2.2- The number of speakers I is known, but not necessarily smaller than the number of microphones J[1].

Assumption 2.3- The impulse responses of all mixing filters are assumed constant during the recordings[2].

B. Channel Estimation

We consider that each recorded signal $\mathbf{x}_j(t)$, $j = 1, \ldots, J$, is a vector of N samples. Let us divide the whole data block into P non-overlapping sub-blocks, such that each sub-block contains $N_P = \lfloor \frac{N}{P} \rfloor$ snapshots. These sub-blocks are indexed by $p = 1, \ldots, P$, and the pth sub-block corresponds to the set of N_P snapshots between instants t_{p-1} and t_p. We denote by $T_P = \frac{N_P}{F_s}$ the duration of each sub-block, where F_s is the sampling frequency. Under this framework, the $J \times J$ autocorrelation matrix $\mathbf{R}_x(f,p) \overset{def}{=} E[\mathbf{x}(f,p)\mathbf{x}^H(f,p)]$ can be written as

$$\mathbf{R}_x(f,p) = \mathbf{H}(f)\mathbf{R}_s(f,p)\mathbf{H}^H(f), \tag{4}$$

where $\mathbf{R}_s(f,p) \overset{def}{=} E[\mathbf{s}(f,p)\mathbf{s}^H(f,p)]$ is the autocorrelation matrix of the speaker signals in the pth sub-block for frequency-bin f. Algorithms that exploit non-stationarity must select T_P such that the successive sub-blocks are uncorrelated. For speech applications, the sub-block duration T_P must be at least 40 ms, as this is generally the lower bound for which speech is considered non-stationary [1]. The statistics are then sufficiently different from one time-lag to another, such that one can simultaneously exploit the P sub-blocks, for a given frequency bin:

$$\left\{ \begin{array}{rcl} \mathbf{R}_x(f,1) & = & \mathbf{H}(f)\mathbf{R}_s(f,1)\mathbf{H}^H(f), \\ \vdots & \vdots & \vdots \\ \mathbf{R}_x(f,P) & = & \mathbf{H}(f)\mathbf{R}_s(f,P)\mathbf{H}^H(f). \end{array} \right. \tag{5}$$

Since we assume mutually uncorrelated speaker signals, we postulate diagonal autocorrelation matrices $\mathbf{R}_s(f,p)$, for $f = 1, \ldots, F$ and $p = 1, \ldots, P$. Estimation of $\mathbf{H}(f)$ thus resumes to a JAD problem for each frequency-bin.

In practice, the exact autocorrelation matrices $\mathbf{R}_x(f,p)$ are unavailable but can be estimated from the samples of $\mathbf{x}(t)$, $t = 1, \ldots, N$. For each sub-block p of N_P samples, we compute the F-point DFT of several consecutive overlapping frames $\mathbf{x}(f,q)$ (each consisting of F temporal samples) with a F-point window (typically a Hanning window). For instance, if α denotes the overlapping factor (e.g. $\alpha = 0.75$), then the number of overlapping frames within each sub-block p is

$$M = \left\lfloor \frac{N_P - N_\alpha}{F - N_\alpha} \right\rfloor, \tag{6}$$

where $N_\alpha = \lfloor \alpha F \rfloor$ is the number of samples in the overlapping segment. The sample autocorrelation matrix estimate, for frequency f and sub-block p, is then given by

$$\begin{array}{rcl} \hat{\mathbf{R}}_x(f,p) & = & \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}(f, k_{p,m}+1 : k_{p,m}+F) \\ & & \mathbf{x}^H(f, k_{p,m}+1 : k_{p,m}+F), \end{array} \tag{7}$$

where $k_{p,m}$ is a super-index that combines p and m as follows

$$k_{p,m} = (p-1)N_P + (m-1)(F - N_\alpha). \tag{8}$$

Typical JAD-based techniques such as [5, 6, 8] require $rank(\mathbf{H}(f)) = I$, for $f = 1, \ldots, F$, therefore they cannot be employed in the under-determined case $J < I$. In the following section, we show that each JAD system (5) can equivalently be written as the PARAFAC decomposition of the third-order tensor $\mathcal{R}_x(f) \in \mathbb{C}^{J \times J \times P}$, built by stacking the P matrices $\left\{\hat{\mathbf{R}}_x(f,1), \ldots, \hat{\mathbf{R}}_x(f,P)\right\}$ one after each other along the third dimension. This PARAFAC-based reformulation was used in [20] for instantaneous mixtures. Its generalization to convolved mixtures implies that the PARAFAC model is now valid for each frequency-bin. One major benefit of the PARAFAC reformulation over the aforementioned JAD techniques is that it does not necessarily require $J \geq I$ for the mixing matrix $\mathbf{H}(f)$ to be unique (up to non-singular scaling and permutation of its columns).

III. Link to the PARAFAC model

A. Reformulation of the problem

In this section, we show that (5) is equivalent to a PARAFAC model. Each element of the tensor $\mathcal{R}_x(f)$ is denoted by $r^{(x)}_{j_1,j_2,p}(f)$, with $j_1 = 1, \ldots, J$, $j_2 = 1, \ldots, J$ and $p = 1, \ldots, P$. The elements of $\mathbf{H}(f)$ are denoted by $h_{j,i}(f)$. We build the $P \times I$ matrix $\mathbf{C}(f)$ whose element on the pth row and ith column, denoted $c_{p,i}(f)$, is the ith diagonal element of $\mathbf{R}_s(f,p)$, i.e., the power spectral density of the ith source within the pth sub-block at frequency-bin f. It follows that the elements $r^{(x)}_{j_1,j_2,p}(f)$ can be written as a sum of triple products

$$r^{(x)}_{j_1,j_2,p}(f) = \sum_{i=1}^{I} h_{j_1,i}(f) c_{p,i}(f) h^*_{j_2,i}(f). \tag{9}$$

Eq. (9) is known as the conjugate-symmetric PARAFAC decomposition of the tensor $\mathcal{R}_x(f)$ and the number of components I is the rank of this tensor [27]. By computing the PARAFAC decomposition of $\mathcal{R}_x(f)$ independently for each frequency-bin, we obtain the entire collection of frequency-domain mixing matrices $\{\mathbf{H}(f), f = 1, \ldots, F\}$ and source power spectra $\{\mathbf{C}(f), f = 1, \ldots, F\}$, up to frequency-dependent permutation and scaling of columns. In the next

[1] If the number of speakers is unknown, it can be estimated as outlined in Section IV-B.

[2] If the mixing environment is varying, the BSS problem has to be solved adaptively. This issue is addressed in Section VII.

section, we discuss the uniqueness conditions for conjugate-symmetric PARAFAC, under which these matrices are identifiable up to the stated indeterminacies.

B. Identifiability

The tensor $\mathcal{R}_x(f)$ is built from elements of the matrices $\mathbf{H}(f)$ and $\mathbf{C}(f)$ combined as in (9). The conjugate-symmetric PARAFAC decomposition of $\mathcal{R}_x(f)$ in Eq. (9) is said to be *essentially unique* if any other matrix pair $\tilde{\mathbf{H}}(f)$ and $\tilde{\mathbf{C}}(f)$ that satisfies (9) is related to $\mathbf{H}(f)$ and $\mathbf{C}(f)$ via

$$\mathbf{H}(f) = \tilde{\mathbf{H}}(f)\mathbf{\Pi}\mathbf{\Lambda}_1, \quad \mathbf{C}(f) = \tilde{\mathbf{C}}(f)\mathbf{\Pi}\mathbf{\Lambda}_2, \tag{10}$$

with $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$ diagonal matrices satisfying $\mathbf{\Lambda}_1\mathbf{\Lambda}_1^*\mathbf{\Lambda}_2 = \mathbf{I}_I$ and $\mathbf{\Pi}$ a permutation matrix. Therefore, the ambiguities of the PARAFAC model are the same as in JAD formulation, i.e., $\mathbf{H}(f)$ and $\mathbf{C}(f)$ are estimated up to arbitrary scaling and permutation of their columns. The way these ambiguities can be corrected will be discussed in Section V.

A first uniqueness result requires the notion of Kruskal-rank of a matrix [27].

Definition 1: The *Kruskal* rank or *k-rank* of a matrix $\mathbf{H}$, denoted by $k_{\mathbf{H}}$, is the maximum number r such that *any* set of r columns of $\mathbf{H}$ forms a linearly independent set.

The following theorem establishes a condition under which essential uniqueness of the conjugate-symmetric PARAFAC decomposition (9) is guaranteed [27,28].

Theorem 1: *The decomposition (9) is essentially unique if*

$$2k_{\mathbf{H}(f)} + k_{\mathbf{C}(f)} \geq 2(I+1). \tag{11}$$

It is worth noting that condition (11) is sufficient but not necessary for identifiability. For a different uniqueness condition, we assume that $I \leq P$. In [29], a relaxed identifiability condition for the conjugate-symmetric PARAFAC model has been derived and is presented in the following theorem.

Theorem 2: *Suppose that the elements of $\mathbf{H}(f)$ and $\mathbf{C}(f)$ are drawn from a jointly continuous distribution. If $I \leq P$ and*

$$\frac{I(I-1)}{2} \leq \frac{J(J-1)}{4}\left(\frac{J(J-1)}{2}+1\right) - \frac{J!}{(J-4)!4!}1_{\{J\geq 4\}}, \tag{12}$$

where

$$1_{\{J\geq 4\}} = \begin{cases} 0, & \text{if } J < 4 \\ 1, & \text{if } J \geq 4 \end{cases},$$

then $\mathbf{H}(f)$ and $\mathbf{C}(f)$ are essentially unique with probability one.

In our context, J corresponds to the number of microphones and I to the number of sources. The following Table gives the upper bound for I such that (12) is satisfied, for different values of J [20]:

J	2	3	4	5	6	7	8
I_{max}	2	4	6	10	15	20	26

From this table, it is clear that the PARAFAC reformulation of the frequency-domain BSS problem allows, in theory, unique identification of the mixing matrices $\mathbf{H}(f)$, for $f = 1, \ldots, F$, even in certain under-determined cases. This is a major advantage over typical JAD techniques, which require $J \geq I$ to solve (5). Note also that invoking uniqueness properties of PARAFAC is a way to prove explicitly that joint-decorrelation of a set of matrices is a sufficient criterion for unique separation.

In the next section, we discuss the batch implementation of the PARAFAC decomposition to separate the sources in the frequency domain, in a static mixing environment.

IV. Batch Implementation

A. Matrix Representation of the Tensor

Most of the algorithms designed to compute the PARAFAC decomposition of a tensor use the different matrix representations of this tensor. In this paper, we will use the following $J^2 \times P$ matrix representation of $\mathcal{R}_x(f)$:

$$[\mathbf{R}_x(f)]_{(j_1-1)J+j_2,p} = [\mathcal{R}_x(f)]_{j_1,j_2,p}, \tag{13}$$

with $j_1 = 1,\ldots,J$, $j_2 = 1,\ldots,J$ and $p = 1,\ldots,P$. By virtue of the conjugate-symmetric PARAFAC model, $\mathbf{R}_x(f)$ is linked to the unknown matrices $\mathbf{H}(f)$ and $\mathbf{C}(f)$ as follows:

$$\mathbf{R}_x(f) = [\mathbf{H}(f) \odot \mathbf{H}^*(f)]\mathbf{C}^T(f). \tag{14}$$

B. Computation of the PARAFAC decomposition

In order to estimate the matrices $\mathbf{H}(f)$ and $\mathbf{C}(f)$ that fit the PARAFAC model of $\mathcal{R}_x(f)$ optimally, an Alternating Least Squares (ALS) algorithm is commonly used. The idea of ALS is to update these matrices in an alternating way at each iteration. We can tentatively ignore symmetry in the model, i.e., treat $\mathbf{H}(f)$ and $\mathbf{H}^H(f)$ as independent variables. Conjugate symmetry of the data in (14) ensures that there is little loss of efficiency in doing so; in the end we can either use one of the two matrix estimates to extract $\mathbf{H}(f)$, or average out the two. We refer to [14,17,30] for further details on ALS. The advantage of ALS is that it works under minimal (model identifiability) conditions; but it can be slow to converge when dealing with ill-conditioned data. An enhanced line search scheme can be inserted in the ALS loop to speed up convergence, as proposed in [31] for the real case and in [32] for the complex case. One can also resort to a Newton-type optimization technique such as the Levenberg-Marquardt algorithm [33]. Note also that the complexity of these algorithms can be significantly reduced by a dimensionality-reduction pre-processing step [34]. Another very efficient algorithm to compute the PARAFAC decomposition was proposed in [35] and used in [20,36]. This algorithm, that we call PARAFAC-SD (for "PARAFAC via Simultaneous Diagonalization") computes the PARAFAC decomposition of a rank-I tensor $\mathcal{R} \in \mathbb{C}^{J_1\times J_2\times J_3}$ via joint-diagonalization of a set of I symmetric matrices of size $I \times I$. It can be applied only under the condition $I \leq min(J_1J_2, J_3)$, where the roles of J_1, J_2 and J_3 can be permuted. This condition is often met in practice, where time is typically the longest dimension J_3 of the observed tensor. Due to its high accuracy and low complexity, the PARAFAC-SD algorithm is a good candidate to solve the BSS problem in this paper. We now briefly describe the principle of this algorithm, as it applies to our particular context. Suppose that $I \leq min(J^2, P)$, which is a realistic assumption for the BSS problem. Let us consider the matrix $\mathbf{R}_x(f) \in \mathbb{C}^{J^2 \times P}$ of Eq. (14). If $rank(\mathbf{H}(f)) = min(I, J)$, then by virtue of a Khatri-Rao product property, $rank(\mathbf{H}(f) \odot \mathbf{H}^*(f)) = I$. Under the assumption $P \geq I$, $\mathbf{C}(f)$ is generically rank-I. As a consequence, $\mathbf{R}_x(f)$ is rank-I and its reduced-size SVD can be written as

$$\mathbf{R}_x(f) = \mathbf{U}(f)\mathbf{\Sigma}(f)\mathbf{V}^H(f), \tag{15}$$

where $\mathbf{U}(f) \in \mathbb{C}^{J^2 \times I}$, $\mathbf{\Sigma}(f) \in \mathbb{R}^{I \times I}$ is diagonal and $\mathbf{V}(f) \in \mathbb{C}^{P \times I}$. Note also that when the number of speakers I is a priori unknown, it can be estimated as the number of significant singular values of $\mathbf{R}_x(f)$, for a given f. The core idea of PARAFAC-SD is to link (14) and (15). Given that $\mathbf{R}_x(f)$ is rank-I, there exists a non-singular matrix $\mathbf{Z}(f) \in \mathbb{C}^{I \times I}$, such that

$$\left\{ \begin{array}{ccc} \mathbf{H}(f) \odot \mathbf{H}^*(f) & = & \mathbf{U}(f)\mathbf{\Sigma}(f)\mathbf{Z}(f) \\ \mathbf{C}^T(f) & = & \mathbf{Z}^{-1}(f)\mathbf{V}^H(f) \end{array} \right. . \tag{16}$$

Estimation of $\mathbf{Z}(f)$ is sufficient to compute the PARAFAC decomposition. Obviously, $\mathbf{C}(f) = \mathbf{V}^*(f)\mathbf{Z}^{-T}(f)$. Also, the columns of $\mathbf{H}(f) \odot \mathbf{H}^*(f)$ are the vectors $\mathbf{h}_i(f) \otimes \mathbf{h}_i^*(f)$, $i = 1, \ldots, I$, which are the vectorized representations of the rank-1 matrices $\mathbf{h}_i(f)\mathbf{h}_i^H(f)$. As a consequence, $\mathbf{h}_i(f)$, $i = 1, \ldots, I$, can be determined, up to a scaling factor, as the left singular vector associated with the largest singular value of the corresponding rank-1 matrix. The key point to finding $\mathbf{Z}(f)$ is to impose that $\mathbf{U}(f)\mathbf{\Sigma}(f)\mathbf{Z}(f)$ has a Khatri-Rao structure. It was shown in [35] for the general un-symmetric PARAFAC decomposition that $\mathbf{Z}(f)$ diagonalizes a set of I symmetric $I \times I$ matrices $\{\mathbf{M}_1(f), \ldots, \mathbf{M}_I(f)\}$ by congruence. For further details on the way these matrices are built, we refer to [20, 35, 36].

This reformulation has two major advantages over classical JAD-based BSS algorithms: i) PARAFAC is uniquely identifiable in certain under-determined cases (see Section III-B), thus *proving* uniqueness of the (estimated) channel matrix, ii) while usual JAD-based techniques jointly diagonalize the initial system of P matrices of size $J \times J$, PARAFAC-SD fully capitalizes on the strong algebraic structure of the PARAFAC model to end up with a smaller JAD problem comprising I matrices of size $I \times I$. The resulting complexity reduction is very significant, even with short signals. Let us consider a simple example with $J = 4$ microphones, $I = 2$ speakers and a short signal split into $P = 12$ epochs. For each frequency, instead of jointly diagonalizing 12 matrices of size 4×4, PARAFAC-SD jointly diagonalizes 2 matrices of size 2×2. With a large FFT length (e.g., 1024 is typical), the complexity advantage over classical JAD methods becomes very pronounced.

The compacted problem for each frequency bin can be solved by any JAD (or PARAFAC) fitting algorithm. The overall accuracy of PARAFAC-SD depends on the algorithm used for this last step. In practice, we will use the extended QZ-iteration [37], as in the original paper [35].

Once the PARAFAC-based separation stage is complete, the scaling and permutation ambiguities have to be corrected. This second stage is addressed in the following section.

V. Scaling and permutation ambiguities

Let $\hat{\mathbf{H}}(f)$ denote an estimate of the matrix $\mathbf{H}(f)$. In the case of perfect estimation, these matrices are linked as follows

$$\hat{\mathbf{H}}(f) = \mathbf{H}(f)\mathbf{D}^{-1}(f)\mathbf{\Pi}^{-1}(f), \tag{17}$$

where $\mathbf{\Pi}(f)$ is an unknown permutation matrix and $\mathbf{D}(f)$ an unknown diagonal matrix. In order to compensate scaling and permutation ambiguities, the task is now to estimate $\mathbf{D}(f)$ and $\mathbf{\Pi}(f)$.

A. Scaling Ambiguity

One possible approach to compensate the scaling ambiguity is the so-called *minimal distortion principle* [26, 38]. We choose $\mathbf{D}(f)$ as

$$\mathbf{D}(f) = diag[\mathbf{Q}\hat{\mathbf{H}}(f)], \tag{18}$$

where $\mathbf{Q} \in \mathbb{R}^{I \times J}$ is a matrix all of whose entries are $1/J$ and $diag(\cdot)$ retains only the diagonal elements and makes the non-diagonal elements zero. This choice of $\mathbf{D}(f)$ can be interpreted as follows. If $\hat{\mathbf{H}}(f)$ is full-column rank for every frequency bin, we can form the demixing matrices $\hat{\mathbf{W}}(f) \overset{def}{=} \hat{\mathbf{H}}^\dagger(f)$, $f = 1, \ldots, F$. The mixing system is characterized at frequency f by the following equation

$$\mathbf{x}(f, q) = \mathbf{H}(f)\mathbf{s}(f, q). \tag{19}$$

If we left-multiply both sides of (19) by $\hat{\mathbf{W}}(f)$, we get

$$\begin{aligned} \hat{\mathbf{s}}(f, q) &\overset{def}{=} \hat{\mathbf{W}}(f)\mathbf{x}(f, q) \\ &= \mathbf{\Pi}(f) diag[\mathbf{Q}\hat{\mathbf{H}}(f)]\mathbf{s}(f, q). \end{aligned} \tag{20}$$

It follows that

$$\hat{s}_i(f, q) = \frac{1}{J}\sum_{j=1}^{J} \hat{h}_{j,i}(f) s_{\Pi(i)}(f, q), \tag{21}$$

where $s_{\Pi(i)}(f, q)$ denotes the ith component of $\mathbf{\Pi}(f)\mathbf{s}(f, q)$. In case of perfect separation, the interpretation of (21) is that the ith output of the BSS algorithm is the average of all observations of the $\Pi(i)$th source across the sensors, when all other sources are switched off. The task is now to estimate the permutation matrices $\mathbf{\Pi}(f)$, $f = 1, \ldots, F$, such that the ith output $\hat{s}_i(f, q)$ in (21) strings together the spectral components originating from the same source $s_{\Pi(i)}(f, q)$ across all frequency bins.

B. Permutation Ambiguity

The spectral alignment is a very challenging problem. If I sources are present, there are $I!$ possible permutations for each frequency bin, which yields a difficult combinatorial problem. Many techniques to solve the permutation problem have been proposed in the literature and we refer to [10] for a survey. Several techniques rely on geometric information, such as estimation of the Direction Of Arrival (DOA), see [26] and references therein. Other techniques rely on the consistency of the filter coeffcients. The latter approach exploits prior knowledge about the mixing filters and the solution can be achieved by requiring the frequency response $\mathbf{H}(f)$ of the mixing filter to be continuous in f [39]. It is also possible to impose smoothness of the demixing filter values in the frequency domain. This is done in [6] by restricting the frequency domain updates of the demixing filter in (2) to have a limited support in the time domain, i.e., $\mathbf{W}(\tau) = 0$ for $\tau > K \ll F$. Restricting the filter length may be problematic in highly reverberant environments where long separation filters are necessary to take all reverberations into account. It is mentioned in [6] that if a long demixing filter length K is needed, one can choose an appropriately large frame size F such that the restriction $K \ll F$ due to the circular convolution approximation still holds. However, large values of F significantly increase the overall complexity. Another category of permutation correction techniques exploits properties of speech signals. One commonly exploited property is the interfrequency correlation of speech signal envelopes [40, 41], which is due to the nature of speech production[3]. For instance, when the talker speaks louder, all spectral components of the signal tend to increase in level, and vice-versa. Based on this idea, several criteria and associated sequential adjustment strategies have been proposed to impose frequency-coupling between adjacent frequency bins, see e.g., [5, 9]. The major drawback of sequential adjustment strategies is *error propagation*, i.e., an error made in the permutation correction at frequency bin f may strongly affect the correction at following frequencies. To avoid this problem, one possible approach is to use a clustering-based method to estimate a frequency-independent reference profile (or centroid) for each separated source, and then permute, for each frequency, the I frequency-dependent profiles such that they all match a different reference profile. This clustering-based idea has been exploited in e.g., [8, 21, 22]. The three key ingredients of these clustering-based techniques are:

i) the definition of the quantities that are clustered, i.e., the source profiles (e.g., signal envelopes, log-power profiles, etc),

ii) the measure used to quantify the matching level between the centroids and the profiles (e.g., correlation, distance, etc), and

iii) the clustering strategy.

In [21], the profile $\hat{\gamma}_i(f,q)$ of a separated signal $\hat{s}_i$ is taken to be its envelope, $\hat{\gamma}_i(f,q) = |\hat{s}_i(f,q)|$. In [22], the profile $\hat{\gamma}_i(f,q)$ is a certain *dominance measure*. In [8], the profile for the ith separated source is defined by its centered log-power spectral density $\hat{\gamma}_i(f,q) = \log[\hat{\mathbf{W}}_{i,:}(f)\mathbf{R}_x(f,q)\hat{\mathbf{W}}^H_{i,:}(f)]$. The length N_f of the profiles is also an important parameter for clustering-based approaches to be accurate, especially for short signals. In practice, the profiles $\hat{\gamma}_i(f,q)$ are computed for overlapping frames over the whole signal. Once the profiles are computed, the task is to compute the centroids and perform clustering. The underlying assumption of clustering-based approaches is that profiles coming from the same source, but at different frequencies, are still more similar than those from other sources. In order to associate each source profile to a centroid for each frequency, one can possibly maximize correlation measures [21, 22] or minimize distance measures [8] across the $I!$ possible permutations for each frequency. At this point, the clustering strategy is crucial. In [8, 21, 22], the centroids and the permutation matrices are updated in an iterative way. For each iteration, the centroids are first updated as the average over all frequencies of the current source profiles. Then, the source profiles are permuted so as to match the current centroids, according to the chosen measure (distance in [8] or correlation in [21, 22]). However, the computation of this measure for the $I!$ permutations and F frequencies at each iteration entails a significant computational cost.

In this section, we propose a more efficient clustering strategy to avoid this problem. Unlike the aforementioned fully iterative methods, the updates of the centroids and permutation matrices are not interleaved, which significantly reduces the complexity. Our scheme can be summarized as follows:

Step 1. Computation of the centroids.

Let us define the $I \times N_f$ matrix $\hat{\mathbf{\Gamma}}(f)$ that collects the I profiles $\hat{\gamma}_i(f)$, $i = 1, \ldots, I$. The $FI \times N_f$ matrix $\underline{\hat{\mathbf{\Gamma}}}$ results from the concatenation of the matrices $\hat{\mathbf{\Gamma}}(f)$, $f = 1, \ldots, F$. Since the profiles have been computed for overlapping frames, $\underline{\hat{\mathbf{\Gamma}}}$ holds a set of FI points varying smoothly with time. The task is now to partition these points into I clusters. This can be done by application of the *k-means* algorithm on $\underline{\hat{\mathbf{\Gamma}}}$, which produces a frequency independent $I \times N_f$ centroid matrix $\mathbf{M} = [\mathbf{m}_1^T, \ldots, \mathbf{m}_I^T]^T$. This centroid matrix is such that the sum over all clusters, of the within-cluster sums of point-to-cluster-centroid distances is minimized[4].

Step 2. Finding the permutation matrices.

For each frequency bin, we now look for the $I \times I$ permutation matrix $\mathbf{\Pi}(f)$ such that $\hat{\mathbf{\Gamma}}(f)\mathbf{\Pi}(f)$ matches $\mathbf{M}$, according to the chosen measure. One possible option [8] is to solve

$$\min_{\mathbf{\Pi}(f)} \phi(f), \ \ f = 1, \ldots, F, \tag{22}$$

where $\phi(f) \overset{def}{=} \left\| \mathbf{M} - \hat{\mathbf{\Gamma}}(f)\mathbf{\Pi}(f) \right\|_F^2$. Another option [21, 22] is to solve

$$\max_{\mathbf{\Pi}(f)} \sum_{i=1}^{I} \rho(\mathbf{m}_i, [\hat{\mathbf{\Gamma}}(f)\mathbf{\Pi}(f)]_{:,i}), \tag{23}$$

where ρ denotes the correlation coefficient. To solve (22) or (23), we compute the exhaustive set of $I!$ measures for each frequency and retain the permutation matrix that corresponds to the best solution[5].

The main feature in our scheme is that only Step 1 is iterative and (22) or (23) is solved only once. This a major

[3] According to the popular source-filter model of speech production, the excitation is filtered through a cascade of second-order oscillators resulting in strong spectral correlation [1].

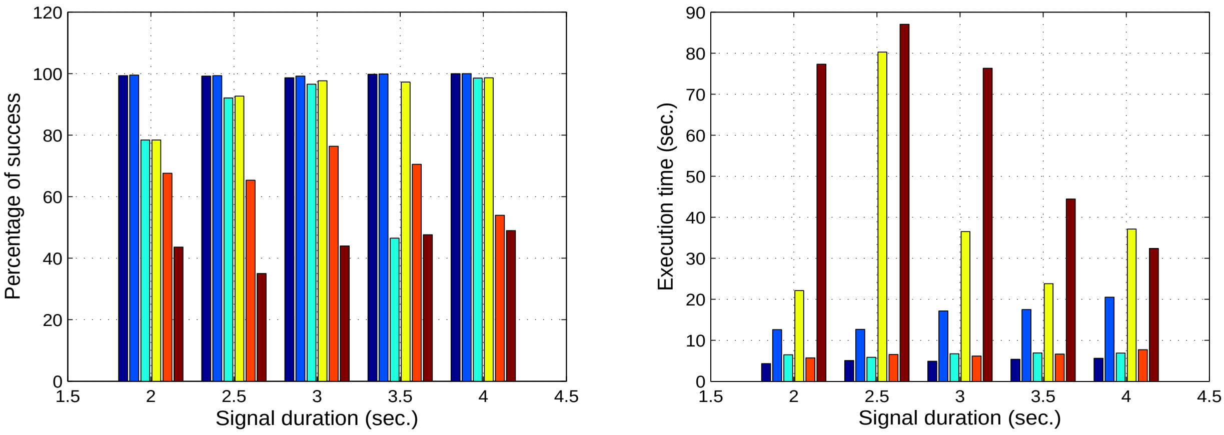
[4] The *k-means* algorithm also produces a list of indices that assigns each of the FI points to one of the I clusters. This list may assign more (or less) than F points to each of the I clusters. We noticed through simulation results that the assignment is however generally very close to F points per cluster which confirms the validity of the aforementioned property of speech signals. Since we have to assign exactly F points to each cluster, we only exploit the centroid matrix $\mathbf{M}$.

[5] To avoid the computation of $I!$ distances at each frequency, one can use a *deflation* approach. For a given frequency, the idea is to associate and remove the best-matching centroid-profile pair from the list of candidates, then repeat the process. This greedy approach is of course suboptimal, but works almost as well in practice.

	Criterion C1	Criteria C2 and C3
Clustering method	Log-power profiles with a distance measure [8]	C2: Dominance profiles with a correlation measure [22]
		C3: Envelope profiles with a correlation measure [21]
iterative	$O(FN_fI^2(I-1)!n)$	$O(FN_fI(I+1)n+FI^2(I-1)!n)$
k-means	$O(FN_fI^2n+FN_fI^2(I-1)!)$	$O(FN_fI^2n+FN_fI^2+FI^2(I-1)!)$

TABLE I

COMPLEXITY OF THE DIFFERENT PERMUTATION CORRECTION SCHEMES. n IS THE NUMBER OF ITERATIONS.



(a) Percentage of success, $I = 5$ sources, $F = 2048$.

(b) CPU time, $I = 5$ sources, $F = 2048$.

Fig. 1. Performance of the three criteria C1, C2 and C3 in solving the permutation problem, combined either with the one-pass *k-means* clustering strategy or the fully iterative strategy. Labels of the six histograms, from left to right: (1) C1 with k-means, (2) C1 iterative, (3) C2 with k-means, (4) C2 iterative, (5) C3 with k-means, (6) C3 iterative.

advantage over the entirely iterative strategies used in [8, 21, 22], where (22) or (23) are solved at each iteration.

C. Comparison between permutation solvers

In this paragraph, we compare the complexity and the performance of the following criteria to solve the permutation problem: (C1) clustering of log-power profiles with a distance measure (22), as proposed in [8], (C2) clustering of dominance-profiles with a correlation measure (23), as proposed in [22], (C3) clustering of envelope-profiles with a correlation measure (23), as proposed in [21]. These criteria are combined either with an entirely iterative clustering strategy, as in their original version, or with the *k-means* approach we proposed. The complexity orders of the different combinations are reported in Table I. It is clear that the clustering strategy that we proposed has a lower complexity than its fully iterative counterpart. This results from the benefit of only estimating the centroids in an iterative way, instead of interleaving updates of centroids and permutation matrices.

In Fig. 1, we compare the performance of the different permutation solvers applied to arbitrarily permuted versions of the *true* source profiles $\mathbf{\Gamma}(f)$, i.e., we simulate the output of a perfect separation stage. The residual frequency-independent permutation is resolved by a column-matching procedure, after which we calculate the number of frequencies for which $\mathbf{\Gamma}(f)$ and $\hat{\mathbf{\Gamma}}(f)$ are perfectly aligned and we compute the percentage of success. The latter is represented by Fig. 1 for $I = 5$ sources. The total execution time is also represented. From this figure, it is clear that clustering the log-power-profiles seems to be a very efficient solution to solve the permutation problem, since its performance index is close to 100 percent, even with 5 sources of 2 seconds only. In comparison, the two other criteria (dominance-profiles and envelope-profiles) are more sensitive to the signal length. As expected, the combination of our *k-means*-based clustering strategy with the three criteria allows a very substantial reduction of the complexity, relative to the entirely iterative approach. Based on these observations, since clustering the log-power profiles with a *k-means*-based strategy offers the best trade-off between complexity and performance, we will use this criterion after the PARAFAC-based separation stage in real BSS situations. In Section VIII-H, we will compare the performance of these different permutation-correction criteria, applied after a PARAFAC-based separation stage, in a real BSS situation.

VI. UNDER-DETERMINED CASE

If $\hat{\mathbf{H}}(f)$ is full-column rank for every frequency bin, separation can be achieved in the frequency-domain by $\hat{\mathbf{s}}(f,q) = \hat{\mathbf{W}}(f)\mathbf{x}(f,q)$, where $\hat{\mathbf{W}}(f) = \hat{\mathbf{H}}^{\dagger}(f)$ is obtained after correction of scaling and permutation ambiguities. The separated sources are then estimated by applying the Inverse DFT to $\{\hat{\mathbf{s}}(f,q), f = 1,\ldots,F\}$. Alternatively, one can first compute the demixing matrix filter $\hat{\mathbf{W}}$ in the time domain, by taking the Inverse DFT of $\left\{\hat{\mathbf{W}}(f), f = 1,\ldots,F\right\}$, after which the deconvolution operation of Eq. (2) may be efficiently computed via an overlap-add procedure. The latter approach will be used in practice.

In the under-determined case, the problem is more difficult. Under the uniqueness conditions reported in Section III-B, PARAFAC allows to identify $\mathbf{H}(f)$ in a unique way, up to scaling and permutation ambiguities. The latter are corrected as explained in Section V. However, the resulting matrix $\hat{\mathbf{H}}(f)$ is not left pseudo-invertible and perfect separation is therefore not possible. In this section, we show that substantial reduction of crosstalk is still possible by using array processing methods, in particular a time-varying version of Capon beamforming. First, we notice that for a sufficiently short sub-block p, the probability that all sources have a high power spectral density simultaneously is low [6] For instance, if $I-J$ sources among I have a long period of pause within sub-block p, the under-determined problem almost resumes to a $J \times J$ determined problem for this sub-block. This suggests that cross-talk reduction should be performed on a per-sub-block basis, to account for variations of cross-talk powers (note that our method automatically adjusts to these variations; it *does not* require activity/pause detection). The task is then to find a set of demixing matrices $\left\{\hat{\mathbf{W}}(f,p),\ f=1,\dots,F,\ p=1,\dots,P\right\}$, such that cross-talk is reduced for each frequency and each sub-block. This can be achieved by Capon beamforming. For a given source i, a given block p and a given frequency f, we look for a $J \times 1$ beamforming vector $\hat{\mathbf{w}}_i(f,p)$ such that

$$\begin{aligned} \hat{s}_i(f,p) &= \hat{\mathbf{w}}_i^H(f,p)\mathbf{x}(f,p) \\ &= \hat{\mathbf{w}}_i^H(f,p)\hat{\mathbf{h}}_i(f)s_i(f,p) \\ &\quad + \textstyle\sum_{k\neq i}\hat{\mathbf{w}}_i^H(f,p)\hat{\mathbf{h}}_k(f)s_k(f,p), \end{aligned} \tag{24}$$

preserves the first term and suppresses the second. Here, $\hat{\mathbf{h}}_i(f)$ denotes the ith column of $\hat{\mathbf{H}}(f)$ after scaling and permutation ambiguities correction. In (24), $\hat{s}_i(f,p)$ results from the sum of a signal of interest and cross-talk signals. The vector $\hat{\mathbf{w}}_i(f,p)$ that minimizes the Signal to Interference Ratio is the Capon beamformer that solves

$$\min_{\hat{\mathbf{w}}_i(f,p)} \ \hat{\mathbf{w}}_i^H(f,p)\mathbf{R}_x(f,p)\hat{\mathbf{w}}_i(f,p) \ \text{ s.t. } \ \hat{\mathbf{w}}_i^H(f,p)\hat{\mathbf{h}}_i(f)=1. \tag{25}$$

The solution of this problem is

$$\hat{\mathbf{w}}_i(f,p) = \frac{\mathbf{R}_x^{-1}(f,p)\hat{\mathbf{h}}_i(f)}{\hat{\mathbf{h}}_i^H(f)\mathbf{R}_x^{-1}(f,p)\hat{\mathbf{h}}_i(f)}. \tag{26}$$

Capon beamforming is then applied at each frequency for each source and each sub-block.

VII. Online Implementation

In the previous sections, we considered a constant mixing environment and we proposed a batch PARAFAC solution of the frequency-domain BSS problem. However, in real-world situations, the mixing system can be considered as constant only over short time intervals, due to speaker mobility, fluctuations in the environment, etc. Online adaptive BSS algorithms are therefore of great interest [3, 42]. In this section, we show that the adaptation of the batch PARAFAC-based BSS technique to the online case can be reduced to the problem of tracking one PARAFAC decomposition for each frequency, for which we have recently proposed efficient adaptive algorithms in [23].

Let us start with Eq. (14), which represents the PARAFAC model of the output autocorrelation tensor $\mathcal{R}_x(f) \in \mathbb{C}^{J\times J\times P}$, in terms of its matrix representation $\mathbf{R}_x(f) \in \mathbb{C}^{J^2\times P}$. If the mixing matrix $\mathbf{H}(f)$ is varying between two successive epochs, it has to be indexed by time and the observed autocorrelation matrix is now

$$\begin{aligned} \mathbf{R}_x(f) \;=\; & \big[(\mathbf{H}(f,1)\odot\mathbf{H}^*(f,1))\,\mathbf{c}^T(f,1),\dots, \\ & (\mathbf{H}(f,P)\odot\mathbf{H}^*(f,P))\,\mathbf{c}^T(f,P)\big], \end{aligned} \tag{27}$$

where $\mathbf{c}^T(f,p)$ is the pth column of $\mathbf{C}^T(f)$. As a consequence, the PARAFAC model, and equivalently the JAD formulation, remain approximately valid only if the mixing-matrix $\mathbf{H}(f,p)$ is almost constant over the P consecutive time-lags. For a sufficiently short time-interval $L_k = [t_k : t_{P+k-1}]$, consisting of P successive time-blocks, we can thus write

$$\mathbf{R}_x(f,L_k) \simeq [\mathbf{H}(f,L_k)\odot\mathbf{H}^*(f,L_k)]\mathbf{C}^T(f,L_k), \tag{28}$$

where $\mathbf{H}(f,L_k) \simeq \mathbf{H}(f,k) \simeq \dots \simeq \mathbf{H}(f,P+k-1)$ and $\mathbf{C}^T(f,L_k) = [\mathbf{c}^T(f,k),\dots,\mathbf{c}^T(f,P+k-1)]$.

The problem can now be summarized as follows:

> *Given estimates of* $\mathbf{H}(f,L_k)$ *and* $\mathbf{C}(f,L_k)$*, estimate* $\mathbf{H}(f,L_{k+1})$ *and* $\mathbf{C}(f,L_{k+1})$ *from the observed matrices* $\mathbf{R}_x(f,L_k)$ *and* $\mathbf{R}_x(f,L_{k+1})$*.*

One possible solution to this problem is to apply a batch PARAFAC algorithm repeatedly on the successive short intervals L_k. Although the batch PARAFAC-SD algorithm proved to be very fast compared to existing JAD techniques, its adaptive version would be very desirable. This is precisely the essence of the PARAFAC-SDT ("PARAFAC via Simultaneous Diagonalization Tracking") algorithm proposed in [23]. PARAFAC-SDT solves (16) adaptively by tracking first the SVD of $\mathbf{R}_x(f)$ before recursively updating $\mathbf{Z}(f)$ and $\mathbf{H}(f)$. For further details on this algorithm, we refer to [23].

In principle, an adaptive permutation solver is also needed to come up with a complete adaptive BSS solution. Thankfully, as we explain in the next section, a side-benefit of tracking using PARAFAC-SDT is that updates are inherently incremental - thus naturally preserving the correct permutation, provided that the adaptive algorithm is properly initialized. Finally, there exist adaptive implementations of Capon beamforming, and these can be easily modified to derive a fully on-line solution that is applicable in under-determined cases as well.

VIII. Simulation Results

A. *Simulation settings*

In this Section, we illustrate the performance of the batch and online PARAFAC-based algorithms developed in this paper. The autocorrelation tensor is computed as explained in Section II-B, with a Hanning window and an overlap coefficient fixed to 75%. In the simulations conducted in this section, we compare our complete solution (PARAFAC-SD separation stage followed by k-means clustering of log-power profiles

[6] This is due to the time-varying spectral characteristics of speech sounds [1], e.g., naturally occurring pauses in speech.

to align the separated spectral components) to the publicly available complete JAD-based batch BSS algorithms proposed in [6] and [5], labeled as "Parra" and "Rahbar", respectively. Parra's algorithm is tested with a demixing-filter of length $F/8$, as in the original paper [6] [7]. Rahbar's algorithm requires the same input parameters as our algorithm, which allows a totally fair comparison. In experiments with $I = 2$ sources and $J = 2$ microphones, we will also compare our algorithm to the JAD-based algorithm of [8], labeled as "Pham", used with the optimal parameters found by preliminary simulations (note that only the implementation for the 2 by 2 case was found on the web for this algorithm).

We have collected a set of nine different signals, consisting of speakers (three females and six males) reading sentences during approximately 30 seconds, with a sampling frequency $F_s = 16$ KHz. These signals are truncated to a chosen length, varying from experiment to experiment. For the comparison between algorithms to be fair, we average the performance over ten random draws of I sources chosen among the nine collected.

In the sequel, performance is assessed in a wide variety of operational scenarios. In Sections VIII-C and VIII-D, we use real recordings of RIRs, resulting from experiments conducted in the context of hearing aid design [43], with 2 microphones. In Section VIII-E, we use the RIRs measured by A. Westner in a conference room [44]. In Sections VIII-F, VIII-G and VIII-H, we use artificial RIRs generated by the method proposed in [45], in order to study the impact of several parameters such as the reverberation time or the location of sources and microphones.

B. Performance Evaluation

From Eq. (2), the separated sources are given by

$$\hat{s}_i(t) = \sum_{j=1}^{J} \mathbf{W}_{ij} \star x_j(t). \tag{29}$$

The output SIR for $\hat{s}_i(t)$ is defined as the ratio of the power of the portion of $\hat{s}_i(t)$ coming from source i, $\hat{s}_{ii}(t)$, to the power from cross-talk signals $\hat{s}_{ik}(t)$ [7]:

$$SIR_i = 10\log \frac{\sum_t \hat{s}_{ii}^2(t)}{\sum_t \sum_{k \neq i} \hat{s}_{ik}^2(t)}. \tag{30}$$

In the experiments of this section, we will convolve speech signals with pre-measured real-world or artificially generated RIRs, so we have access to the microphone signals $x_{ji}(t)$, $j = 1, \ldots, J$, recorded when only the ith source is present. Therefore, we calculate the SIR for source i as[8]

$$SIR_i = 10\log \frac{\sum_t \left(\sum_{j=1}^{J} \mathbf{W}_{ij} \star x_{ji}(t) \right)^2}{\sum_t \sum_{k \neq i} \left(\sum_{j=1}^{J} \mathbf{W}_{ij} \star x_{jk}(t) \right)^2}. \tag{31}$$

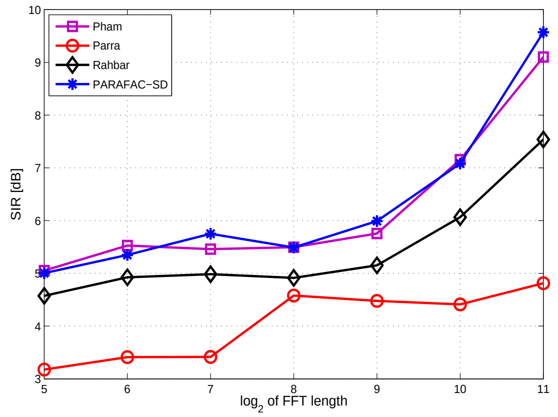


Fig. 2. Impact of FFT length, 2 by 2 case, $T_P = 0.25$ sec, $T_{60} = 130$ ms.

We will use the SIR averaged over all sources as a single overall performance measure. The input SIR, i.e., the SIR obtained without any processing, will also be given as a baseline.

C. Experiment 1: two by two case

In this first experiment (Figs. 2 and 3), we compare the different batch algorithms with $I = 2$ sources and $J = 2$ microphones. We have used real recordings of RIRs, resulting from experiments conducted in the context of hearing aid design [43]. The chosen room is a semi-reverberant classroom with dimensions $17'10''$ by $32'9''$ by $8'8''$ (named PC335 in the database). The reverberation time T_{60} is around 130 ms. These recordings allow to choose between different positions of the speakers on a circle around the microphones by selection of angles between $0°$ and $338°$. The radius of the circle is $3'$. The signal duration is fixed to 10 s and the duration of each sub-block is $T_P = 0.25$ s, i.e., the recordings are partitioned in $P = 40$ segments. Performance is averaged over 5 different pairs of positions, one source being fixed at $0°$ while the second is successively positioned at $45°$, $90°$, $135°$, $180°$ and $225°$. As mentioned previously, performance is also averaged over ten random pairs of sources.

In Fig. 2, we illustrate the impact of the FFT length F on the output SIR. The average input SIR was -2.1 dB in this experiment. It turns out that PARAFAC-SD and Pham's algorithms achieve similar SIR and outperform Rahbar's and Parra's techniques. Comparison of execution times (not shown here) revealed that PARAFAC-SD was between 1 and 2 decades faster than the three other batch algorithms.

In Fig. 3, we test the four algorithms on truncated recordings, whose duration is varying from 2 s to 10 s. The FFT length is fixed to $F = 2048$. Figs. 3(a) and 3(b) represent evolution of the output SIR and execution time, respectively. For a short signal (between 2 s and 4 s), our method substantially outperforms Parra's and Rahbar's techniques and slightly outperforms Pham's method. This results from the combination of a fast and accurate PARAFAC-based separation stage, followed by a fast and accurate permutation

[7] Preliminary results with other filter lengths have shown that $F/8$ offers the best performance in most (but not all) of the cases considered in this section.

[8] In the under-determined case where Capon beamforming is used on a per-sub-block basis, the inverse filter varies across sub-blocks. In this case SIR_i is computed in a similar way, except that $\hat{s}_{ii}^2(t)$ and $\hat{s}_{ik}^2(t)$ in (30) are built by concatenation of their successively estimated sub-blocks.

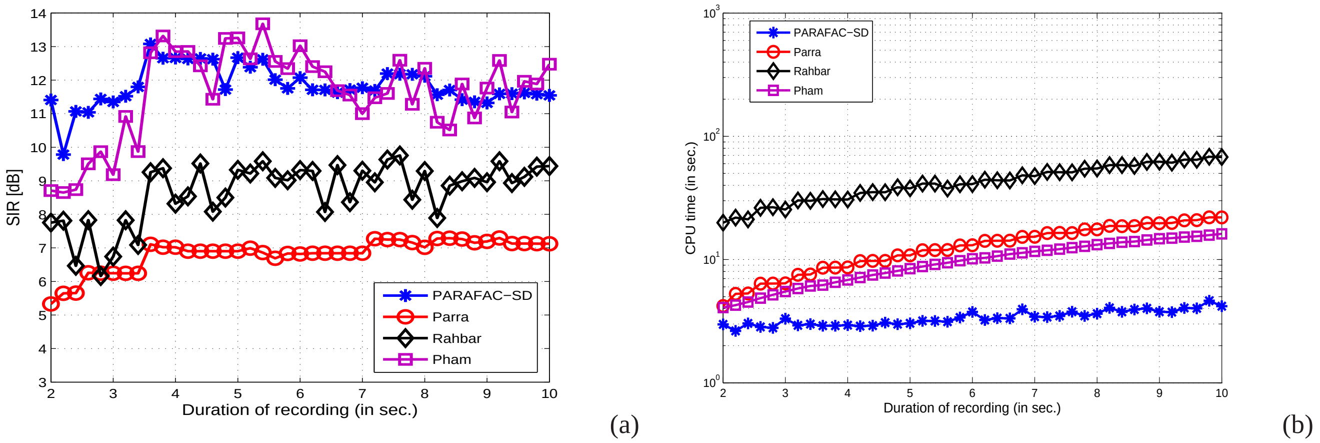


Fig. 3. Impact of signal duration, 2 by 2 case, $F = 2048$, $T_P = 0.25$ s, $T_{60} = 130$ ms. (a) Evolution of SIR. (b) Evolution of execution time.

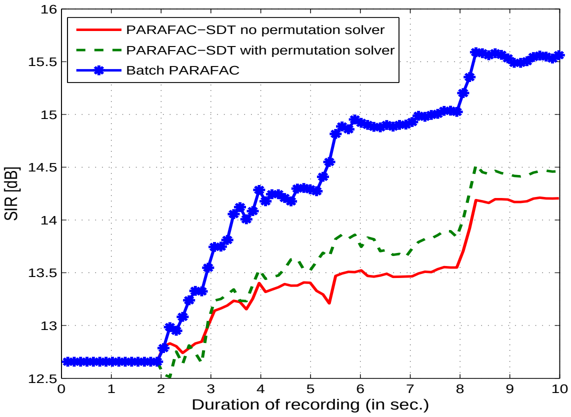


Fig. 4. Performance of PARAFAC-SDT algorithm in the 2 by 2 case. $F = 1024$, $T_P = 0.128$ s, $P_{init} = 15$ ($\simeq 2$ s), $T_{60} = 70$ ms. Average Input $SIR = -1.82$ dB. Static environment. Speakers positioned at 0° and 90°. Evolution of SIR vs. signal duration (average over 10 random pairs of sources). Comparison between batch PARAFAC and online PARAFAC-SDT (with or without solving the permutation problem at each step of the online mode).

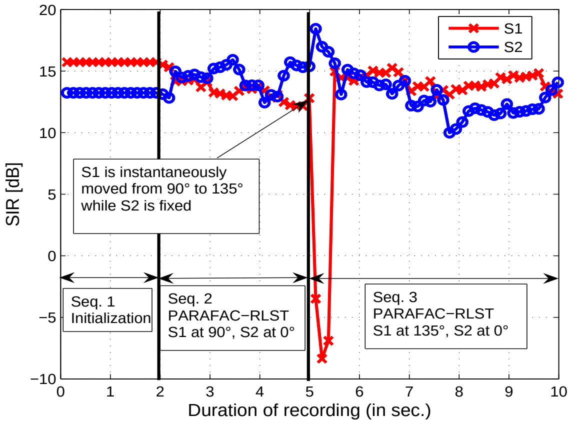


Fig. 5. Performance of PARAFAC-SDT algorithm in the 2 by 2 case. $F = 1024$, $T_P = 0.128$ s, $P_{init} = 15$ ($\simeq 2$ s), $T_{60} = 70$ ms. Varying environment. Evolution of output SIR for each speaker. Sequence 1: initialization with batch PARAFAC-SD on $P_{init} = 15$ sub-blocks, speakers positioned at 0° and 90°. Sequence 2: online mode, positions are the same as in Seq. 1. Sequence 3: speaker 2 keeps the same position, while speaker 1 is moved instantaneously. Average Input $SIR = -1.65$ dB for Sequences 1 and 2, and -2.48 dB for Sequence 3.

correction scheme, which proved to work well even with short signals (see Section V-C). From 4 seconds, PARAFAC-SD and Pham's algorithms perform similarly, and outperform Rahbar's and Parra's algorithms. Note that PARAFAC-SD is always faster than the three other algorithms, and becomes much faster when the signal duration increases. The signal duration has little impact on the execution time of the PARAFAC-based separation stage since the latter *always* reduces the dimension of the problem to a set of I matrices to jointly diagonalize (the number of matrices to diagonalize is reduced from $P = 40$ to $P = I = 2$ in this experiment). Of course, the execution time of the global solution shown in Fig. 3(b) increases with time, since the permutation correction scheme has to cluster profiles of increasing length.

D. Experiment 2: Adaptive PARAFAC

In this second experiment (Figs. 4 and 5), we illustrate the performance of the online PARAFAC-SDT algorithm. We used room PC323c from the same database [43], with $I = 2$ sources and $J = 2$ microphones. The reverberation time T_{60} is around 70 ms. The FFT length is fixed to $F = 1024$ and the epoch duration to $T_P = 0.128$ s.

In Fig. 4, the mixing environment is constant. We compare the performance of the batch PARAFAC-SD algorithm applied repeatedly on signals of increasing length to that of its online counterpart (PARAFAC-SDT), used with a sliding exponentially decaying window of length 10 sub-blocks and a forgetting factor equal to 0.8 (see [23] for details on this algorithm). We have plotted the evolution of the SIR averaged over both users and ten random pairs of sources. For a given sub-block p, the SIR of a given user is computed by Eq. (31), where $\mathbf{W}_{ij}$ is substituted by its estimate $\hat{\mathbf{W}}_{ij}(p)$ for this block and $x_{ji}(t)$ and $x_{jk}(t)$ consist of all available samples (i.e., pN_P samples) of the recorded signals up to the pth block. PARAFAC-SDT is initialized with the mixing matrix estimated by batch PARAFAC-SD applied on the first $P_{init} = 15$ sub-blocks (i.e., approximately 2 seconds). Then, PARAFAC-SDT is combined with one of the two following options for the rest of the recording: (O1) the permutation problem is globally resolved for each new block (after the recursive updates) by taking into account all previous blocks; or (O2) it is never solved in online mode. From Fig. 4, it is clear that both options yield similar performance. The reason is that PARAFAC-SDT recursively updates the new matrices explicitly as a function of the old estimates, such that the tracking stage does not introduce new arbitrary permutations. Consequently, since the frequency-dependent permutation problem is well solved in the initialization step (this is due to the effectiveness of the permutation correction scheme for short signals), it is not necessary to solve it again in online mode. From this first observation, we deduce that the small performance gap (around 1 dB only) between batch PARAFAC-SD and its online version results from the separation stage only. On the other hand, PARAFAC-SDT has a much lower complexity than its batch counterpart [23]; it was on average 20 times faster than PARAFAC-SD in this experiment.

In Fig. 5, we illustrate the tracking capability of PARAFAC-SDT. During the first 5 seconds, the sources are fixed at 90° and 0°, respectively. After 5 seconds, the first source is instantaneously moved from 90° to 135°, while the second source is kept fixed. The SIR of each speaker was computed as follows. In the first sequence (initialization) of $P_{init} = 15$ blocks, we applied the batch PARAFAC-SD algorithm, and the SIR of each user resulting from (31) is replicated P_{init} times in the figure. In the second sequence (online mode between $t = 2$ s and $t = 5$ s), both users have the same position as in the first sequence, and we compute the SIR as before. In the third sequence, SIR for the second speaker (who remains in the same position) is computed on the whole data up to present time, whereas SIR for the first speaker (who moves instantaneously at $t = 5$ sec) is only computed over samples corresponding to $t > 5$ s. The key point is that the update of the demixing filter for this speaker does not exploit the benefit of a "good" initialization (with batch PARAFAC-SD), since the mixing-environment has been instantaneously changed. We observe that after 4 sub-blocks (about half a second), the SIR of the first speaker reaches a level close to its initial value, which illustrates the very good tracking capability of the PARAFAC-SDT algorithm. Note that this good tracking capability is also illustrated in [23], in a completely different context (tracking the trajectories of multiple targets in a MIMO radar system).

E. Experiment 3: Highly reverberant environment

Although the database used in the first two experiments provides real world RIRs recordings, it is limited to $J = 2$ sensors only, since it was built in the context of hearing aid design [43]. In this third experiment, we use the RIRs measured by A. Westner in a conference room of size 3.5 m×7 m×3 m, with 8 microphones [44]. The duration of these RIRs is 750 ms, such that the full room acoustics is captured, and the reverberation time T_{60} is around 300 ms, which characterizes a highly reverberant environment. The duration of the sources is fixed to 10 seconds and performance is averaged over 10 random draws of the sources.

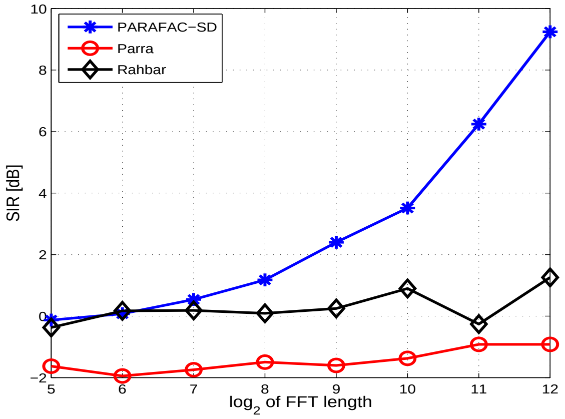


Fig. 6. Westner's RIRs recordings. Impact of FFT length. $I = 3$ sources, $J = 6$ microphones, $T_P = 0.5$ s. $T_{60} = 300$ ms. Input $SIR = -2.8$ dB.

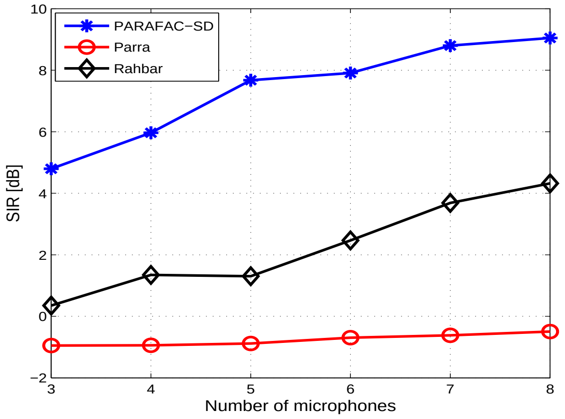


Fig. 7. Westner's RIRs recordings. Impact of the number of microphones. $I = 3$ sources, $T_P = 0.5$ s, $F = 4096$, $T_{60} = 300$ ms. Input SIR between -3.5 dB and -1.46 dB, depending on the value of J.

In Fig. 6, we illustrate the impact of the FFT length with $I = 3$ sources and $J = 6$ sensors. As observed in the 2 by 2 case, PARAFAC-SD outperforms Parra's and Rahbar's techniques in terms of output SIR. In terms of execution time, PARAFAC-SD was approximately 10 times faster than Parra's algorithm and 100 times faster than Rahbar's algorithm.

In Fig. 7, F is fixed to 4096 and we illustrate the impact of the number of microphones, with $I = 3$ sources. Contrary to Parra's and Rahbar's techniques, PARAFAC-SD achieves "satisfactory" separation quality with only 3 microphones. When J increases, the quality of separation improves for the three algorithms but PARAFAC-SD yields the best output SIR.

F. Experiment 4: Under-determined case

In this fourth experiment (Fig. 8), we consider under-determined cases and we illustrate the performance of PARAFAC-SD algorithm followed by Capon beamforming, as described in Section VI. The sources have 10 seconds duration

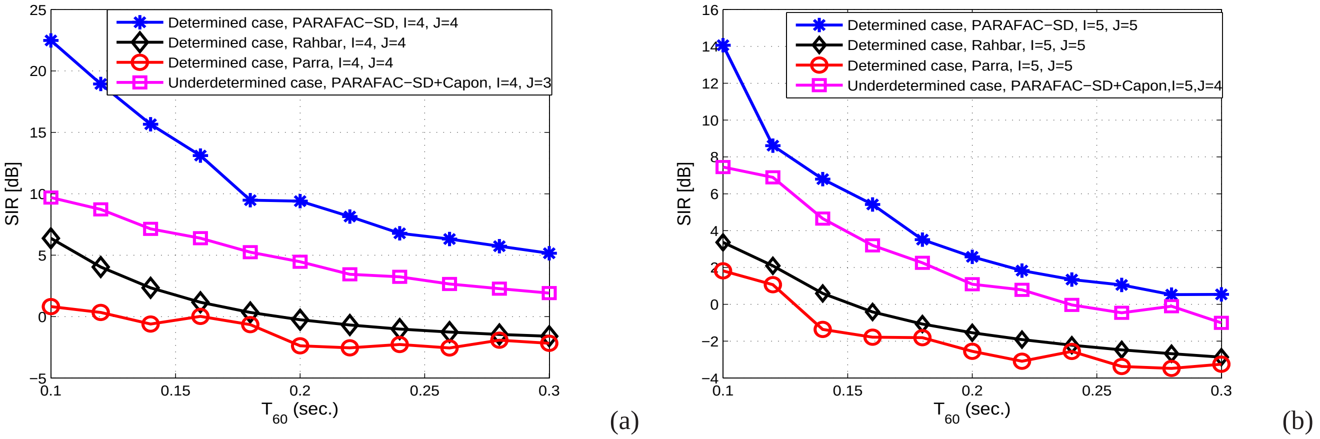


Fig. 8. Performance of PARAFAC-SD followed by time-varying Capon beamforming in the under-determined case, $F = 2048$, $T_P = 0.256$ s. Comparison with the determined case with PARAFAC-SD, Parra's or Rahbar's algorithms. Input SIR between -2.02 dB and -4.84 dB, depending on the value of T_{60}. (a) $I = 4$ sources. (b) $I = 5$ sources.

and they are convolved with artificial RIRs, generated by the method proposed in [45][9]. Artificial RIRs generators allow to test BSS algorithms in various situations, since the dimensions of the room, the locations of the sources and microphones and the reverberation time can be freely chosen. In this experiment, the dimensions of the chosen room are 5 m$\times 5$ m$\times 2.3$ m. The RIRs are generated for $I = 5$ sources and $J = 5$ microphones. The x and z coordinates of the 5 sources are fixed to 2 and 1.6, respectively, while the y coordinates are $\{1, 1.5, 2, 2.5, 3\}$. The x and z coordinates of the 5 sensors are fixed to 3 and 1.6, respectively, while the y coordinates are $\{1, 1.4, 1.8, 2.2, 2.6\}$. F is fixed to 2048 and T_P to 0.5 s. The performance is averaged over 10 random draws of the sources.

In Fig. 8(a), only the first 4 sources have been mixed and we represent the evolution of the SIR averaged over all sources as a funtion of the reverberation time T_{60}, in the two following situations:

i) The first 4 microphones are used. In this exactly determined case, the estimated mixing matrix is invertible and the same demixing filter $\mathbf{W}_{ij}$ is therefore used for all sub-blocks. The performance of PARAFAC-SD, Parra's and Rahbar's algorithms is plotted.

ii) The first 3 microphones only are used. In this under-determined case, the mixing matrix is first estimated by PARAFAC, after which the demixing filters $\mathbf{W}_{ij}(p)$ are estimated by Capon beamforming for each sub-block.

In Fig. 8(b), we proceed similarly to compare the 5 by 5 exactly determined case to the 5 by 4 under-determined case.

As a conclusion, though the separation quality naturally decreases with an increasing reverberation time, PARAFAC-SD (followed by Capon beamforming) performs very well in the under-determined case. In particular, it significantly outperforms Parra's and Rahbar's techniques even when the latter two are given the benefit of using one more microphone, thus operating in the exactly determined regime. This is indicative of the strengths of the proposed approach. It is also worth noticing that the gap between the under-determined and the exactly determined cases can be quite small for PARAFAC-SD + Capon, see Fig. 8(b). Additional experiments for challenging under-determined cases can be found at `http://www.telecom.tuc.gr/~nikos/BSS_Nikos.html`.

G. Experiment 5: variable source and microphone positions

In this fifth experiment (Fig. 9), we compare the performance of the three batch algorithms as a function of the locations of the sources and the microphones. The number of sources is $I = 2$ and the number of microphones $J = 6$. Performance is averaged over 10 random draws of the sources. As in the previous section, we use artificial RIR's [45]. The size of the room is 12 m $\times$ 9 m $\times$ 3 m and the reverberation time is fixed to $T_{60} = 200$ ms. The signals have 5 seconds duration.

In a first scenario (Fig. 9(a)), we observe the impact of the distance between the microphones. PARAFAC-SD significantly outperforms Parra's and Rahbar's algorithms. When the distance between microphones increases, the performance of the three techniques improves. This was expected, since increasing this distance decreases the correlation between the different RIRs, which in turn, makes the simultaneous diagonalization problem better conditioned.

In a second scenario (Fig. 9(b)), we proceed similarly, but this time we vary the distance between the sources. We observe that the separation performance improves when this distance increases, up to a certain point. Notice also that PARAFAC-SD works very well (giving SIR of 12 dB) when the sources are only 20 cm apart.

In a third scenario (Fig. 9(c)), we observe the impact of the distance between sources and sensors. Again, PARAFAC-SD significantly outperforms Parra's and Rahbar's algorithms. When the sources are getting closer to the microphone array, the performance of the three algorithms improves. This was expected since the convolutive mixing problem is then getting closer to a simpler instantaneous mixing problem (one dominant direct path with high energy, relatively to the reflected paths).

[9] `http://home.tiscali.nl/ehabets/rir_generator.html`

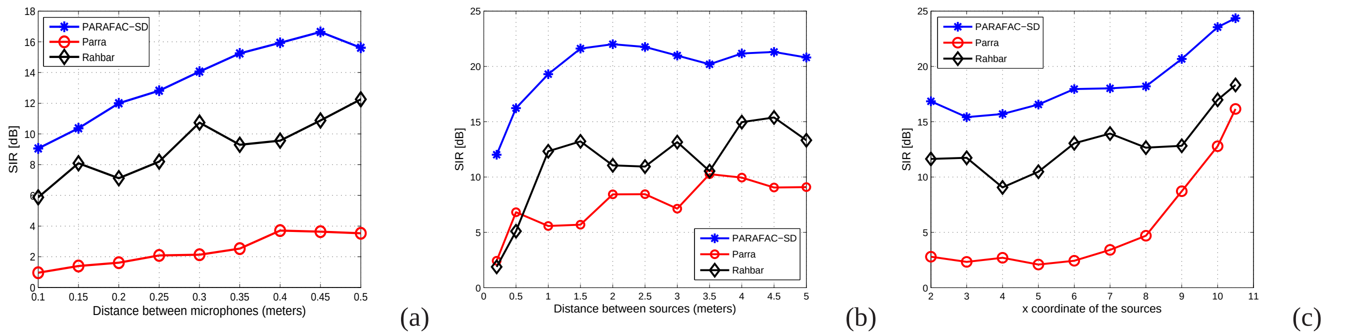


Fig. 9. Impact of sources and sensors locations. $I = 2$ sources, $J = 6$ microphones. $F = 2048$, $T_P = 0.256$ s. Room of size 12 m × 9 m × 3 m, $T_{60} = 200$ ms.
(a) Impact of inter-microphone distance. Sources: $\{(2, 1, 1.6), (2, 2, 1.6)\}$. Microphones: $\{(11, (j-1)d_m + 1, 1.6)\}_{j=1,\ldots,6}$, with distance d_m varying from 0.1 m to 0.5 m. Average Input $SIR = -2.56$ dB.
(b) Impact of inter-source distance. Sources: $\{(5, 1, 1.6), (5, 1 + y_s, 1.6)\}$, with y_s varying from 0.2 to 5. Microphones: $\{(8, 0.3(j-1)+1, 1.6)\}_{j=1,\ldots,6}$. Average Input $SIR = -3.02$ dB.
(c) Impact of the distance between sources and microphones. Sources: $\{(x_s, 1, 1.6), (x_s, 2, 1.6)\}$, with x_s varying from 2 to 10.5. Microphones: $\{(11, 0.3(j-1)+1, 1.6)\}_{j=1,\ldots,6}$. Average Input $SIR = -2.72$ dB.

H. Experiment 6: Comparison of permutation criteria

In this last experiment (Fig. 10), we apply the different permutation-correction criteria proposed in Section V-B after a PARAFAC-SD separation stage, for varying reverberation times. The room has the same dimensions as in the previous experiment. The number of sources is $I = 3$, and the number of microphones $J = 8$. The signal duration is 5 s. The coordinates of the sources are $(10, 1, 1.6)$, $(10, 2, 1.6)$ and $(10, 3, 1.6)$. The x and z coordinates of the 8 sensors are fixed to 11 and 1.6, respectively, while the y coordinates are $\{1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4\}$. It can be observed that criteria C1 (clustering log-power profiles with a distance measure) and C2 (dominance profiles with a correlation measure) yield similar performance and outperform criterion C3 (envelope profiles with a correlation measure). This confirms the observations made in Section V-C. Computation of C1 and C2 via the *k-means*-based approach we proposed yields performance that is similar to the entirely iterative clustering strategy, but the *k-means* strategy has a far lower complexity (see Table I).

Several additional experiments (including challenging under-determined cases and speech-music mixtures) are available at `http://www.telecom.tuc.gr/~nikos/BSS_Nikos.html`.

IX. Conclusion

In this paper, we have proposed a PARAFAC-based approach to solve the BSS problem for convolutive speech mixtures in the frequency domain. Our approach is very competitive, since it provides better separation performance at much lower complexity relative to the state-of-art. These benefits come from combining a fast and accurate PARAFAC algorithm for the separation stage, with an efficient frequency-dependent permutation correction scheme.

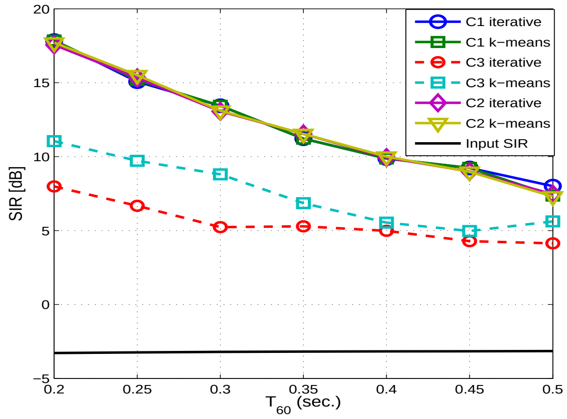


Fig. 10. Comparison between several permutation correction criteria after the same PARAFAC-SD separation stage. $I = 3$ sources, $J = 8$ microphones, $F = 2048$ and $T_P = 0.256$ s.

Contrary to earlier work in blind speech separation, the link to PARAFAC allows estimation of the mixing matrix in under-determined cases - there is *proof* of identifiability. Although perfect separation is not even theoretically possible in under-determined cases, we have shown that exploitation of the estimated (fat) channel matrix together with time-varying Capon beamforming affords significant cross-talk reduction. We have also constructed an adaptive solution that features good tracking performance and low complexity. Finally, extensive experiments with realistic and measured data have been conducted to corroborate our findings, including a performance comparison with two BSS algorithms from the state of the art, in a large variety of mixing scenarios.

References

[1] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*. Prentice-Hall, 1978.

[2] D.-T. Pham and J.-F. Cardoso, “Blind separation of instantaneous mixtures of non-stationary sources,” in *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA’00)*, Helsinki, Finland, 2000, pp. 187–193.

[3] R. Aichner, H. Buchner, S. Araki, and S. Makino, “On-line time-domain blind source separation of nonstationary convolved signals,” in *Proc. Int.*

Workshop on Indep. Comp. Analysis and Blind Sig. Separation (ICA'03), 2003, pp. 987–992.

[4] A. Gorokhov and P. Loubaton, "Subspace Based Techniques for Second Order Blind Separation of Convolutive Mixtures with Temporally Correlated Sources," *IEEE Trans. on Circuit and Systems*, vol. 44, no. 9, pp. 813–820, 1997.

[5] K. Rahbar and J.-P. Reilly, "A frequency domain method for blind source separation of convolutive audio mixtures," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 832–844, 2005. [Online]. Available: http://www.ece.mcmaster.ca/~reilly/kamran/id18.htm

[6] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, 2000. [Online]. Available: http://ida.first.fhg.de/~harmeli/download/download_convbss.html

[7] C. Servière and D.-T. Pham, "Permutation correction in the frequency domain in blind separation of speech mixtures," *EURASIP J. Appl. Signal Proc.*, no. 1, pp. 1–16, 2006.

[8] D.-T. Pham, C. Servière, and H. Boumaraf, "Blind separation of speech mixtures based on nonstationarity," in *Proc. of ISSPA'03*, vol. 2, 2003, pp. 73–76. [Online]. Available: http://www.lis.inpg.fr/pages_perso/bliss/toolboxes/bssaudio-demo.tar.gz

[9] N. Mitianoudis and M. Davies, "Audio source separation of convolutive mixtures," *IEEE Trans. on Speech and Audio Processing*, vol. 11, no. 5, pp. 489–497, 2003.

[10] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods," in *Springer Handbook of Speech Processing*. Springer Press, 2007.

[11] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signals," *Radar and Signal Processing, IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370, 1993.

[12] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. on Signal Processing*, vol. 50, no. 7, pp. 1545–1553, 2002.

[13] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second order statistics," *IEEE Trans. on Signal Processing*, vol. 45, no. 2, pp. 434–444, fev. 1997.

[14] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and Conditions for an 'explanatory' Multi-mode Factor Analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[15] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*. Chichester, U.K.: John Wiley and Sons, 2004.

[16] P. Kroonenberg, *Applied Multiway Data Analysis*. Wiley Series in Probability and Statistics, 2008.

[17] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC Receivers for DS-CDMA Systems," *IEEE Trans. on Signal Proc.*, vol. 48, pp. 810–823, 2000.

[18] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel Factor Analysis in Sensor Array Processing," *IEEE Trans. on Signal Proc.*, vol. 48, pp. 2377–2388, 2000.

[19] P. Comon, "Blind Identification and Source Separation in 2 × 3 Underdetermined Mixtures," *IEEE Trans. on Signal Proc.*, vol. 52, no. 1, pp. 11–22, 2004.

[20] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. on Signal Proc.*, vol. 56, no. 3, pp. 1096–1105, 2008.

[21] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A Robust and Precise Method for Solving the Permutation Problem of Frequency-Domain Blind Source Separation," *IEEE Trans. on Speech and Audio Processing*, vol. 12, no. 5, pp. 530–538, 2004.

[22] H. Sawada, S. Araki, and S. Makino, "MLSP 2007 Data Analysis Competition: Frequency-Domain Blind Source Separation for Convolutive Mixtures of Speech/Audio Signals," in *Proc. MLSP2007*, 2007, pp. 45–50.

[23] D. Nion and N. D. Sidiropoulos, "Adaptive Algorithms to Track the PARAFAC Decomposition of a Third-Order Tensor," *IEEE Trans. Signal Proc.*, vol. 57, no. 6, pp. 2299–2310, 2009.

[24] K. N. Mokios, N. D. Sidiropoulos, and A. Potamianos, "Blind Speech Separation Using Parafac Analysis and Integer Least Squares," in *Proc. ICASSP 06*, vol. 5, 2006, pp. 73–76.

[25] K. N. Mokios, A. Potamianos, and N. D. Sidiropoulos, "On the Effectiveness of PARAFAC-Based Estimation for Blind Speech Separation," in *Proc. ICASSP 08*, 2008, pp. 153–156.

[26] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Frequency-Domain Blind Source Separation of Many Speech Signals Using Near-Field and Far-Field Models," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–13, 2006.

[27] J. B. Kruskal, "Three-way Arrays: Rank and Uniqueness of Trilinear Decompositions, with Application to Arithmetic Complexity and Statistics," *Linear Algebra Appl.*, vol. 18, pp. 95–138, 1977.

[28] A. Stegeman and N. D. Sidiropoulos, "On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition," *Lin. Alg. Appl.*, vol. 420, pp. 540–552, 2007.

[29] A. Stegeman, J. ten Berge, and L. De Lathauwer, "Sufficient conditions for uniqueness in Candecomp/Parafac and Indscal with random component matrices," *Psychometrika*, vol. 71, pp. 219–229, 2006.

[30] R. Bro, "PARAFAC: Tutorial and Applications," *Chemom. Intell. Lab. Syst.*, vol. 38, pp. 149–171, 1997.

[31] M. Rajih and P. Comon, "Enhanced Line Search: A Novel Method to Accelerate PARAFAC," in *Proc. Eusipco'05*, 2005.

[32] D. Nion and L. De Lathauwer, "An Enhanced Line Search Scheme for Complex-Valued Tensor Decompositions. Application in DS-CDMA," *Signal Proc.*, vol. 88, no. 3, pp. 749–755, 2008.

[33] G. Tomasi and R. Bro, "A Comparison of Algorithms for Fitting the PARAFAC Model," *Comp. Stat. Data Anal.*, vol. 50, pp. 1700–1734, 2006.

[34] L. De Lathauwer and J. Vandewalle, "Dimensionality Reduction in Higher-Order Signal Processing and Rank-$(r_1, r_2, \ldots, r_n)$ Reduction in Multilinear Algebra," *Lin. Alg. Appl.*, Special Issue on *Linear Algebra in Signal and Image Processing*, vol. 391, pp. 31–55, Nov. 2004.

[35] L. De Lathauwer, "A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization," *SIAM J. Matrix Anal. Appl.*, vol. 28, no. 3, pp. 642–666, 2006.

[36] L. De Lathauwer and J. Castaing, "Tensor-Based Techniques for the Blind Separation of DS-CDMA signals," *Signal Processing*, Special Issue *Tensor Signal Processing*, vol. 87, no. 2, pp. 322–336, 2007.

[37] A.-J. van der Veen and A. Paulraj, "An Analytical Constant Modulus Algorithm," *IEEE Trans. Signal Proc.*, vol. 44, pp. 1136–1155, 1996.

[38] K. Matsuoka and S. Nakashima, "Minimal distortion principle for blind source separation," in *Proc. Int. Workshop on Indep. Comp. Analysis and Blind Sig. Separation (ICA'01)*, 2001, pp. 722–727.

[39] D.-T. Pham, C. Servière, and H. Boumaraf, "Blind separation of convolutive audio mixtures using nonstationarity," in *Proc. Int. Workshop on Indep. Comp. Analysis and Blind Sig. Separation (ICA'03)*, 2003, pp. 981–986.

[40] J. Anemüller and B. Kollmeier, "Amplitude Modulation Decorrelation for Convolutive Blind Source Separation," in *Proc. Int. Workshop on Indep. Comp. Analysis and Blind Sig. Separation (ICA '00)*, 2000, pp. 215–220.

[41] N. Murata, S. Ikeda, and A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals," *Neurocomputing*, vol. 41, pp. 1–24, Oct. 2001.

[42] L. Parra and C. Spence, "On-line convolutive source separation of non-stationary signals," *J. of VLSI Signal Processing*, vol. 26, no. 1-2, Aug. 2000.

[43] L. Trainor, R. Sonnadara, K. Wiklund, J. Bondy, S. Gupta, S. Becker, I.-C. Bruce, and S. Haykin, "Development of a flexible, realistic hearing in noise test environment (R-HINT-E)," *Signal Processing*, vol. 84, no. 2, pp. 299–309, Feb. 2004. [Online]. Available: http://trainorlab.mcmaster.ca/ahs/rhinte.htm

[44] A. Westner and J. V. M. Bove, "Blind Separation of Real World Audio Signals Using Overdetermined Mixtures," in *Proc. ICA'99*, 1999. [Online]. Available: http://sound.media.mit.edu/ica-bench/

[45] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, April 1979.