JOINT INDSCAL DECOMPOSITION MEETS BLIND SOURCE SEPARATION

Thanh Trung Le[⋆,‡]*, Karim Abed-Meraim*[‡]*, Philippe Ravier*[‡]*, Olivier Buttelli*[‡]*, Ales Holobar*[†]

[⋆] AVITECH Institute, VNU University of Engineering and Technology, Hanoi, Vietnam
[‡] PRISME Laboratory, University of Orléans, Orléans, France
[†] Institute of Computer Science, University of Maribor, Maribor, Slovenia

ABSTRACT

This paper introduces TenSOFO, a novel tensor-based method specifically designed for blind source separation (BSS). TenSOFO presents a new efficient alternating direction method of multipliers framework, allowing for simultaneous decomposition of two symmetric third-order tensors under the individual differences in scaling (INDSCAL) format. By establishing a fundamental link between joint INDSCAL decomposition and BSS using second and fourth order statistics, TenSOFO proves to be effective for BSS. The performance of TenSOFO is evaluated in both joint INDSCAL decomposition and BSS tasks, showcasing its remarkable accuracy and potential applications.

Index Terms— Blind source separation, second-order statistics, fourth-order statistics, tensor decomposition, INDSCAL.

1. INTRODUCTION

Let's consider the basic blind source separation (BSS) model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \quad t = 0, 1, 2, \ldots, T-1, \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^M$ represents the data observation, $\mathbf{A} \in \mathbb{R}^{M\times R}$ is the mixing matrix, $\mathbf{s}(t) \in \mathbb{R}^R$ is the source vector, $\mathbf{n}(t) \in \mathbb{R}^R$ denotes the noise. The main objective of BSS is to estimate the source signals and/or the mixing matrix from the set of data observations $\{\mathbf{x}(t)\}_{t\geq 0}^{T-1}$.

The literature offers many methods proposed for BSS. For a comprehensive overview, readers are referred to good references such as [1–3]. Notably, with the advances in tensor decomposition (TD), several algorithms have leveraged tensor formats to reformulate BSS models, leading to a promising application of TD in BSS [3]. A pioneering tensor-based BSS algorithm was introduced by De Lathauwer *et al.* in [4] wherein the higher-order SVD model was deployed to perform BSS. They further proposed an independent component analysis algorithm by means of simultaneous tensor diagonalization in [5]. Subsequently, in [6], a connection between canonical polyadic (CP) decomposition and joint (simultaneous) diagonalization was established, paving the way for various CP-based BSS methods. Some notable examples of CP-based BSS methods include tensor-pICA [7], FOOBI [8], SOBIUM [9], CP-VDM [10], PARAFAC-SD [11], and DC-CPD [12]. Another tensor approach in BSS is the block component analysis or block term decomposition (BTD) [13, 14]. Specifically, tensorization techniques like Hankelization [15], Löwnerization [16], and segmentation [17] have been developed to enable the use of BTD for BSS on certain classes of source signals such as exponential and polynomials.

In BSS tasks, statistical properties of data play a crucial role [18]. Many BSS methods have effectively utilized second-order (SO) and/or fourth-order (FO) statistics for source separation, as demonstrated in [19–25], among others. SO statistics offer insights into the correlation and linear relationships among observed signals, while FO statistics can capture the non-Gaussian nature of the sources and higher-order dependencies among the data observations. Most existing methods, however, either focus solely on one type of data statistics (e.g., SO or FO) or partially exploit their information. Consequently, our objective is to fully integrate both SO and FO statistics with tensor analysis for BSS. This approach is expected to provide a more comprehensive set of statistical features and leverage the benefits of tensor representation, leading to enhanced robustness and accuracy in performance.

In this paper, we contribute to the literature on blind source separation by introducing a new tensor method that effectively utilizes both SO and FO statistics. Our method is based on a joint (simultaneous) analysis called individual differences in scaling (INDSCAL), which is a symmetric variant of CP for third-order tensors with symmetry in two modes [26,27]. INDSCAL offers improved interpretability as compared to the classical CP model for BSS, which uses SO and FO represented by symmetric matrices (covariance) and tensors (quadricovariance), respectively. We propose an effective alternating direction method of multipliers (ADMM)-based joint INDSCAL decomposition of two symmetric third-order tensors. Additionally, we establish a link between SO-FO based BSS and joint INDSCAL decomposition, allowing us to effectively apply the proposed INDSCAL method for BSS tasks.

Notations: Scalars, vectors, matrices, and tensors are denoted using lowercase, boldface lowercase, boldface capital, and bold calligraphic letters, respectively. Symbols $\circ, \odot, \otimes, \circledast$, and $\boxplus$ represent the outer, Khatri-Rao, Kronecker, Hadamard products, and tensor concatenation, respectively. We use the operators $\mathrm{vec}(\cdot)$ for vectorization, $\mathrm{diag}(\cdot)$ for diagonalization, $\mathrm{mat}(\cdot)$ for matricization, and $\mathrm{length}(\cdot)$ for calculating the number of entries in a vector, matrix, or tensor. Transpose is represented as $(\cdot)^\mathsf{T}$, pseudo-inverse as $(\cdot)^\#$, and the norm as $\|\cdot\|$. We denote the trace inner product as $\langle \mathbf{X}, \mathbf{Y} \rangle = \mathrm{tr}(\mathbf{X}^\mathsf{T}\mathbf{Y})$. The mode-$n$ unfolding matrix of $\boldsymbol{\mathcal{X}}$ is denoted as $\mathbf{X}_{(n)}$. Lastly, the CP/PARAFAC decomposition is denoted by $[\![\cdot]\!]$.

2. PRELIMINARIES

2.1. BSS Using Data Statistics

Let's consider a data vector $\mathbf{u}(t) \in \mathbb{R}^I$ of zero mean. We can analyze its second-order (SO) and fourth-order (FO) statistics

This work was supported by the European Pro-Athena program under grant No. 20-GURE-0012. Ales Holobar was supported by the Slovenian Research and Innovation Agency (Programme No. P2-0041).

using the covariance matrix $\mathbf{R}^{\mathbf{u}} \in \mathbb{R}^{I\times I}$ and the quadricovariance tensor $\boldsymbol{\mathcal{C}}^{\mathbf{u}} \in \mathbb{R}^{I\times I\times I\times I}$, respectively. These statistics are defined as follows:

$$\mathbf{R}^{\mathbf{u}}(t,\tau) = \mathbb{E}\big\{\mathbf{u}(t)\mathbf{u}(t-\tau)^{\mathsf{T}}\big\}, \tag{2}$$

$$\boldsymbol{\mathcal{C}}^{\mathbf{u}}_{ijkl}(t,\{\boldsymbol{\tau}\}) = \mathrm{Cum}\big\{u_i(t), u_j(t-\tau_1), u_k(t-\tau_2), u_l(t-\tau_3)\big\}, \tag{3}$$

for the time lags τ and $\boldsymbol{\tau} \equiv \{\tau_1,\tau_2,\tau_3\}$, where $u_i(t)$ is the i-th entry of $\mathbf{u}(t)$. In this work, we express the quadricovariance tensor $\boldsymbol{\mathcal{C}}^{\mathbf{u}}$ as a matrix $\mathbf{C}^{\mathbf{u}}$ of size $I^2\times I^2$, whose elements are defined as $\mathbf{C}^{\mathbf{u}}_{(i-1)I+j,(k-1)I+l} = \boldsymbol{\mathcal{C}}^{\mathbf{u}}_{ijkl}$. From (1), we obtain

$$\mathbf{R}^{\mathbf{x}}(t,\tau) = \mathbf{A}\mathbf{R}^{\mathbf{s}}(t,\tau)\mathbf{A}^{\mathsf{T}} + \mathbf{R}^{\mathbf{n}}(t,\tau) \tag{4}$$

$$\mathbf{C}^{\mathbf{x}}(t,\{\boldsymbol{\tau}\}) = \mathbf{A}_{\odot 2}\mathbf{C}^{\mathbf{s}}(t,\{\boldsymbol{\tau}\})\mathbf{A}_{\odot 2}^{\mathsf{T}} + \mathbf{C}^{\mathbf{n}}(t,\{\boldsymbol{\tau}\}). \tag{5}$$

where $\mathbf{A}_{\odot 2} = \mathbf{A}\odot\mathbf{A}$. Given a set of statistics $\{\mathbf{R}^{\mathbf{x}}(t,\tau_n)\}_{n=1}^{N_1}$ and $\{\mathbf{C}^{\mathbf{x}}(t,\{\boldsymbol{\tau}_n\})\}_{n=1}^{N_2}$, we want to identify the mixing $\mathbf{A}$.

To facilitate the development of our algorithm in the next section, we assume that the additive noise is white, Gaussian distributed and the underlying sources are stationary, non-Gaussian and mutually statistically independent, while being individually correlated for different lags.

2.2. INDSCAL

The INdividual Differences in SCALing (INDSCAL) represents a special variant of CP/PARAFAC decomposition that enables the factorization of symmetric tensors [26, 27]. Under the INDSCAL model, a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times I\times K}$ with elements satisfying $x_{ijk} = x_{jik}$ for all i,j,k can be decomposed into two factors $\mathbf{A}\in\mathbb{R}^{I\times R}$ and $\mathbf{C}\in\mathbb{R}^{K\times R}$ (R being the tensor rank) as follows

$$\boldsymbol{\mathcal{X}} \triangleq [\![\mathbf{A},\mathbf{A},\mathbf{C}]\!] = \textstyle\sum_{r=1}^{R}\mathbf{a}_r\circ\mathbf{a}_r\circ\mathbf{c}_r, \tag{6}$$

where $\mathbf{a}_r$ and $\mathbf{c}_r$ are the r-th columns of $\mathbf{A}$ and $\mathbf{C}$, respectively. Its computation typically follows the same iterative procedure to computing the classical CP decomposition (i.e., CP-ALS) [27]. In CP-ALS, the two "$\mathbf{A}$" matrices are treated as separate factors, denoted as $\mathbf{A}_{\mathrm{L}}$ and $\mathbf{A}_{\mathrm{R}}$ (for left and right, respectively), and they are updated independently without an explicit constraint enforcing their equality. Despite starting with different initial estimates, the inherent symmetry of the data together with the essential uniqueness of CP decomposition eventually lead the two "$\mathbf{A}$" matrices to converge, up to a scaling factor. Being a special case of CP, the uniqueness of INDSCAL is also guaranteed under mild conditions [28].

2.3. ADMM

The Alternating Direction Method of Multipliers (ADMM) is an effective primal-dual optimization framework designed to deal with convex constrained problems of the form [29]

$$\min_{\mathbf{x},\mathbf{y}} f(\mathbf{x}) + g(\mathbf{y}) \quad \text{subject to} \quad \mathbf{z}(\mathbf{x},\mathbf{y}) = \mathbf{c}. \tag{7}$$

The augmented Lagrangian corresponding to (7) is given by

$$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\mu}) = f(\mathbf{x}) + g(\mathbf{y}) + \frac{\rho}{2}\big\|\mathbf{c}-\mathbf{z}(\mathbf{x},\mathbf{y})\big\|_F^2 + \boldsymbol{\mu}^{\mathsf{T}}(\mathbf{c}-\mathbf{z}(\mathbf{x},\mathbf{y})), \tag{8}$$

where $\rho>0$ is a regularized parameter and $\boldsymbol{\mu}$ is the dual variable. ADMM relies on the duality theory for convex optimization, where the objective is to minimize the augmented Lagrangian w.r.t. $\mathbf{x}$, $\mathbf{y}$ and a fixed $\boldsymbol{\mu}$. Conversely, the dual function $h(\boldsymbol{\mu}) = \min_{\mathbf{x},\mathbf{y}}\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\mu})$ should be maximized w.r.t. $\boldsymbol{\mu}$. Consequently, ADMM performs an alternation between minimizing $\mathcal{L}(\cdot)$ w.r.t. $\mathbf{x}$ and $\mathbf{y}$ and employing gradient ascent to maximize $h(\boldsymbol{\mu})$. In the next section, we adapt this ADMM framework to compute a joint INDSCAL decomposition, enabling blind source separation (BSS) using data statistics.

3. PROPOSED METHOD

In this section, we first present a fundamental connection between joint INDSCAL decomposition and BSS. Subsequently, we introduce a novel tensor method for BSS named TenSOFO (where Ten, SO, FO, refer to Tensors, Second Order statistics, and Fourth Order statistics, respectively), which effectively leverages both SO and FO statistics.

3.1. SO-FO Based BSS As Joint INDSCAL Decomposition

Under the assumptions stated in Section 2.1, we obtain

$$\mathbf{R}^{\mathbf{s}}(t,\tau) = \mathrm{diag}\big\{\sigma_1^2(\tau),\sigma_2^2(\tau),\ldots,\sigma_R^2(\tau)\big\}, \tag{9}$$

$$\mathbf{C}^{\mathbf{s}}(t,\{\boldsymbol{\tau}\}) = \mathrm{diag}\big\{\kappa_1(\{\boldsymbol{\tau}\}),\kappa_2(\{\boldsymbol{\tau}\}),\ldots,\kappa_R(\{\boldsymbol{\tau}\})\big\}, \tag{10}$$

where $\sigma_r^2(\tau) = \mathbb{E}\{s_r(t)s_r(t-\tau)\}$ and $\kappa_r(\{\boldsymbol{\tau}\}) = \mathrm{Cum}\{s_r(t), s_r(t-\tau_1), s_r(t-\tau_2), s_r(t-\tau_3)\}$. Accordingly, if we construct two tensors $\boldsymbol{\mathcal{R}}\in\mathbb{R}^{M\times M\times N_1}$ and $\boldsymbol{\mathcal{C}}\in\mathbb{R}^{M^2\times M^2\times N_2}$ from the set of N_1 matrices[1] $\{\mathbf{R}^{\mathbf{x}}(t,\tau_n)\}_{n=1}^{N_1}$ and N_2 matrices $\{\mathbf{C}^{\mathbf{x}}(t,\boldsymbol{\tau}_n)\}_{n=1}^{N_2}$ as follows

$$\boldsymbol{\mathcal{R}} = \mathbf{R}^{\mathbf{x}}(t,\tau_1)\boxplus\mathbf{R}^{\mathbf{x}}(t,\tau_2)\boxplus\cdots\boxplus\mathbf{R}^{\mathbf{x}}(t,\tau_{N_1}), \tag{11}$$

$$\boldsymbol{\mathcal{C}} = \mathbf{C}^{\mathbf{x}}(t,\{\boldsymbol{\tau}_1\})\boxplus\mathbf{C}^{\mathbf{x}}(t,\{\boldsymbol{\tau}_2\})\boxplus\cdots\boxplus\mathbf{C}^{\mathbf{x}}(t,\{\boldsymbol{\tau}_{N_2}\}), \tag{12}$$

then they admit the following INDSCAL factorization [3]

$$\boldsymbol{\mathcal{R}} \approx [\![\mathbf{A},\mathbf{A},\boldsymbol{\Sigma}]\!] \quad\text{and}\quad \boldsymbol{\mathcal{C}} \approx [\![\mathbf{A}_{\odot 2},\mathbf{A}_{\odot 2},\mathbf{K}]\!]. \tag{13}$$

Here, $\boldsymbol{\Sigma}\in\mathbb{R}^{N_1\times R}$ is a matrix whose the (n,r)-th element represents the autocorrelation $\sigma_r^2(\tau_n)$, while the (n,r)-th element of $\mathbf{K}\in\mathbb{R}^{N_2\times R}$ corresponds to $\kappa_r(\{\boldsymbol{\tau}_n\})$. Consequently, through the joint INDSCAL decomposition of both $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ in (13), we directly estimate the mixing matrix $\mathbf{A}$. In next subsection, we present an efficient optimization framework to simultaneously decompose $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$.

3.2. Optimization Framework

The joint INDSCAL decomposition of $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ in (13) can be obtained by solving the following constrained minimization

$$\min f(\mathbf{B},\mathbf{K}) + g(\mathbf{A},\boldsymbol{\Sigma}) \quad \text{subject to} \quad \mathbf{B} = \mathbf{A}_{\odot 2}, \tag{14}$$

where $f(\mathbf{B},\mathbf{K}) = \|\boldsymbol{\mathcal{C}} - [\![\mathbf{B},\mathbf{B},\mathbf{K}]\!]\|_F^2$ and $g(\mathbf{A},\boldsymbol{\Sigma}) = \|\boldsymbol{\mathcal{R}} - [\![\mathbf{A},\mathbf{A},\boldsymbol{\Sigma}]\!]\|_F^2$. Here, (14) can be expressed in the ADMM form, and thus, we can construct the corresponding augmented Lagrangian function with a parameter $\rho_{\bigcirc}$ as follows

$$\mathcal{L}_{\bigcirc}(\mathbf{B},\mathbf{K},\mathbf{A},\boldsymbol{\Sigma},\mathbf{U}) = f(\mathbf{B},\mathbf{K}) + g(\mathbf{A},\boldsymbol{\Sigma}) + \frac{\rho_{\bigcirc}}{2}\big\|\mathbf{B}-\mathbf{A}_{\odot 2}+\mathbf{U}\big\|_F^2 - \frac{\rho_{\bigcirc}}{2}\big\|\mathbf{U}\big\|_F^2, \tag{15}$$

where $\mathbf{U}\in\mathbb{R}^{M^2\times R}$ is the (scaled) dual variable. The proposed ADMM solver can be summarized as follows

[1] Since the noise is white which means that $\mathbf{R}^{\mathbf{n}}(t,\tau) = \mathbf{0}$ for $\tau\neq 0$, we usually select non-zero lags to get rid of the noise term in (4).

while stopping criteria are not met **do**

$$\{\mathbf{B}^{(k)},\mathbf{K}^{(k)}\} = \underset{\mathbf{B},\mathbf{K}}{\operatorname{argmin}} \Big\{ f(\mathbf{B},\mathbf{K}) + \frac{\rho_{\circ}}{2}\big\|\mathbf{B} - \mathbf{A}_{\odot 2}^{(k-1)} + \mathbf{U}^{(k-1)}\big\|_F^2 \Big\} \quad (16a)$$

$$\{\mathbf{A}^{(k)},\mathbf{\Sigma}^{(k)}\} = \underset{\mathbf{A},\mathbf{\Sigma}}{\operatorname{argmin}} \Big\{ g(\mathbf{A},\mathbf{\Sigma}) + \frac{\rho_{\circ}}{2}\big\|\mathbf{B}^{(k)} - \mathbf{A}_{\odot 2} + \mathbf{U}^{(k-1)}\big\|_F^2 \Big\} \quad (16b)$$

$$\mathbf{U}^{(k)} = \mathbf{U}^{(k-1)} + \mathbf{B}^{(k)} - \mathbf{A}_{\odot 2}^{(k)} \quad (16c)$$

$k = k+1$

end

Updates of $\mathbf{B}^{(k)}$ and $\mathbf{K}^{(k)}$: Minimization (16a) is equivalent to the following constrained optimization

$$\underset{\mathbf{B}_{\mathrm{L}},\mathbf{B}_{\mathrm{R}},\mathbf{K}}{\operatorname{argmin}} \big\|\mathcal{C} - [\![\mathbf{B}_{\mathrm{L}},\mathbf{B}_{\mathrm{R}},\mathbf{K}]\!]\big\|_F^2 + \frac{\rho_{\circ}}{2}\big\|\mathbf{B}_{\mathrm{L}} - \mathbf{A}_{\odot 2}^{(k-1)} + \mathbf{U}^{(k-1)}\big\|_F^2 \quad \text{subject to } \mathbf{B}_{\mathrm{L}} = \mathbf{B}_{\mathrm{R}}, \quad (17)$$

where the two "$\mathbf{B}$" matrices in $f(\mathbf{B},\mathbf{K})$ are considered as separate loading factors, denoted as $\mathbf{B}_{\mathrm{L}}$ and $\mathbf{B}_{\mathrm{R}}$ for the left and right, respectively. The corresponding augmented Lagrangian function is expressed as follows

$$\mathcal{L}_{\mathrm{B}}(\mathbf{B}_{\mathrm{L}},\mathbf{B}_{\mathrm{R}},\mathbf{K},\mathbf{D}) = \big\|\mathcal{C} - [\![\mathbf{B}_{\mathrm{L}},\mathbf{B}_{\mathrm{R}},\mathbf{K}]\!]\big\|_F^2 - \frac{\rho_{\mathrm{B}}}{2}\big\|\mathbf{D}\big\|_F^2 + \frac{\rho_{\circ}}{2}\big\|\mathbf{B}_{\mathrm{L}} - \mathbf{A}_{\odot 2}^{(k-1)} + \mathbf{U}^{(k-1)}\big\|_F^2 + \frac{\rho_{\mathrm{B}}}{2}\big\|\mathbf{B}_{\mathrm{L}} - \mathbf{B}_{\mathrm{R}} + \mathbf{D}\big\|_F^2. \quad (18)$$

Here, $\rho_{\mathrm{B}} > 0$ is a regularized parameter, and $\mathbf{D} \in \mathbb{R}^{M^2 \times R}$ denotes the (scaled) dual variable. To find the optimal solution of (18), the optimization process involves an iterative loop, with the i-th iteration step as follows:

$$\mathbf{K}^{(k,i)} = \mathbf{C}_{(3)}\Big(\mathbf{B}_{\mathrm{L}}^{(k,i-1)} \odot \mathbf{B}_{\mathrm{R}}^{(k,i-1)}\Big)\Big(\big[(\mathbf{B}_{\mathrm{L}}^{(k,i-1)})^{\top}\mathbf{B}_{\mathrm{L}}^{(k,i-1)}\big] \circledast \big[(\mathbf{B}_{\mathrm{R}}^{(k,i-1)})^{\top}\mathbf{B}_{\mathrm{R}}^{(k,i-1)}\big]\Big)^{\#}, \quad (19a)$$

$$\mathbf{B}_{\mathrm{L}}^{(k,i)} = \Big(\mathbf{C}_{(1)}\mathbf{P}_{\mathrm{R}}^{(k,i-1)} + \rho_{\circ}\big(\mathbf{A}_{\odot 2}^{(k-1)} - \mathbf{U}^{(k-1)}\big) + \rho_{\mathrm{B}}\big(\mathbf{B}_{\mathrm{R}}^{(k,i-1)} - \mathbf{D}^{(k,i-1)}\big)\Big)\Big(\big(\mathbf{P}_{\mathrm{R}}^{(k,i-1)}\big)^{\top}\mathbf{P}_{\mathrm{R}}^{(k,i-1)} + (\rho_{\circ}+\rho_{\mathrm{B}})\mathbf{I}_R\Big)^{-1}, \quad (19b)$$

$$\mathbf{B}_{\mathrm{R}}^{(k,i)} = \Big(\mathbf{C}_{(2)}\mathbf{P}_{\mathrm{L}}^{(k,i-1)} + \rho_{\mathrm{B}}\big(\mathbf{B}_{\mathrm{L}}^{(k,i-1)} - \mathbf{D}^{(k,i-1)}\big)\Big)\Big(\big(\mathbf{P}_{\mathrm{L}}^{(k,i-1)}\big)^{\top}\mathbf{P}_{\mathrm{L}}^{(k,i-1)} + \rho_{\mathrm{B}}\mathbf{I}_R\Big)^{-1}, \quad (19c)$$

$$\mathbf{D}^{(k,i)} = \mathbf{D}^{(k,i-1)} + \mathbf{B}_{\mathrm{L}}^{(k,i)} - \mathbf{B}_{\mathrm{R}}^{(k,i)}, \quad (19d)$$

where $\mathbf{P}_{\mathrm{L}}^{(k,i-1)} = \mathbf{K}^{(k,i)} \odot \mathbf{B}_{\mathrm{L}}^{(k,i-1)}$ and $\mathbf{P}_{\mathrm{R}}^{(k,i)} = \mathbf{K}^{(k,i)} \odot \mathbf{B}_{\mathrm{R}}^{(k,i-1)}$. At the initial step ($i = 0$), we set $\mathbf{K}^{(k,0)} = \mathbf{K}^{(k-1)}$ and $\mathbf{B}_{\mathrm{L}}^{(k,0)} = \mathbf{B}_{\mathrm{R}}^{(k,0)} = \mathbf{B}^{(k-1)}$. The iterative procedure (19) continues until convergence or until stopping criteria are met after I_{stop} iterations. We align the two "$\mathbf{B}$" matrices as follows

$$\{\mathbf{\Pi}_{\mathbf{B}}^{(k)},\mathbf{\Lambda}_{\mathbf{B}}^{(k)}\} = \underset{\mathbf{\Pi}_{\mathbf{B}},\mathbf{\Lambda}_{\mathbf{B}}}{\operatorname{argmin}} \big\|\mathbf{B}_{\mathrm{L}}^{(k,I_{\mathrm{stop}})} - \mathbf{B}_{\mathrm{R}}^{(k,I_{\mathrm{stop}})}\mathbf{\Pi}_{\mathbf{B}}\mathbf{\Lambda}_{\mathbf{B}}\big\|_F, \quad (20)$$

where $\mathbf{\Pi}_{\mathbf{B}}$ and $\mathbf{\Lambda}_{\mathbf{B}}$ are a permutation matrix and a diagonal matrix, respectively. At the end, we take $\mathbf{\Lambda}_{\mathbf{B}}^{(k)} = \mathbf{\Lambda}_{\mathbf{B}}^{(k,I_{\mathrm{stop}})}$ and $\mathbf{B}^{(k)} = 0.5(\mathbf{B}_{\mathrm{L}}^{(k,I_{\mathrm{stop}})} + \mathbf{B}_{\mathrm{R}}^{(k,I_{\mathrm{stop}})}\mathbf{\Pi}_{\mathbf{B}}^{(k)}\mathbf{\Lambda}_{\mathbf{B}}^{(k)})$.

Updates of $\mathbf{A}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$: We also recast (16b) into the following ADMM form

$$\underset{\mathbf{A}_{\mathrm{L}},\mathbf{A}_{\mathrm{R}},\mathbf{\Sigma}}{\operatorname{argmin}} \big\|\mathcal{R} - [\![\mathbf{A}_{\mathrm{L}},\mathbf{A}_{\mathrm{R}},\mathbf{\Sigma}]\!]\big\|_F^2 + \frac{\rho_{\circ}}{2}\big\|\mathbf{B}^{(k)} - \mathbf{A}_{\odot 2} + \mathbf{U}^{(k-1)}\big\|_F^2 \quad \text{subject to } \mathbf{A}_{\mathrm{L}} = \mathbf{A}_{\mathrm{R}}. \quad (21)$$

In a way similar to (19), we employ an iterative procedure to update $\mathbf{A}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$. Starting at $j = 0$, we initialize $\mathbf{\Sigma}^{(k,0)} = \mathbf{\Sigma}^{(k-1)}$, and obtain $\mathbf{A}_{\mathrm{L}}^{(k,0)}$ and $\mathbf{A}_{\mathrm{R}}^{(k,0)}$ from $\mathbf{A}_{\odot 2}^{(k,0)}$ which is computed as follows

$$\mathbf{A}_{\odot 2}^{(k,0)} = \Big(\mathbf{R}_{(3)}^{\top}\mathbf{\Sigma}^{(k,0)} + \rho_{\circ}(\mathbf{B}^{(k)} + \mathbf{U}^{(k-1)})\Big)\Big(\big(\mathbf{\Sigma}^{(k,0)}\big)^{\top}\mathbf{\Sigma}^{(k,0)} + \rho_{\circ}\mathbf{I}\Big)^{-1}. \quad (22)$$

Specifically, we know that $\mathbf{A}_{\odot 2}^{(k,0)} = \mathbf{A}_{\mathrm{L}}^{(k,0)} \odot \mathbf{A}_{\mathrm{R}}^{(k,0)}$ leads to

$$\mathbf{A}_{\odot 2}^{(k,0)}(:,r) = \mathrm{vec}\big\{\mathbf{A}_{\mathrm{L}}^{(k,0)}(:,r)\mathbf{A}_{\mathrm{R}}^{(k,0)}(:,r)^{\top}\big\}, 1 \le r \le R. \quad (23)$$

We can obtain the (normalized) r-th column of $\mathbf{A}_{\mathrm{L}}^{(k,0)}$ and $\mathbf{A}_{\mathrm{R}}^{(k,0)}$ from the most dominant left and right singular vectors of $\mathrm{reshape}\big\{\mathbf{A}_{\odot 2}^{(k,0)}(:,r),[M,M]\big\}$, respectively. At the j-th iteration, we denote $\mathbf{F}^{(k)} = \mathbf{B}^{(k)} + \mathbf{U}^{(k-1)}$, $\mathbf{Q}_{\mathrm{L}}^{(k,j-1)} = \mathbf{\Sigma}^{(k,j)} \odot \mathbf{A}_{\mathrm{L}}^{(k,j-1)}$, $\mathbf{Q}_{\mathrm{R}}^{(k,j-1)} = \mathbf{\Sigma}^{(k,j)} \odot \mathbf{A}_{\mathrm{R}}^{(k,j-1)}$, $\mathbf{G}_{\mathrm{L}}^{(k,j-1)} = (\mathbf{I}_R \odot \mathbf{A}_{\mathrm{L}}^{(k,j-1)}) \otimes \mathbf{I}_M$, $\mathbf{G}_{\mathrm{R}}^{(k,j-1)} = (\mathbf{I}_R \odot \mathbf{A}_{\mathrm{R}}^{(k,j-1)}) \otimes \mathbf{I}_M$, $\mathbf{T}_{\mathrm{L}}^{(k,j-1)} = \mathrm{reshape}\{\mathbf{G}_{\mathrm{L}}^{(k,j-1)}\mathrm{vec}\{\mathbf{F}^{(k)}\},[M,R]\}$, $\mathbf{T}_{\mathrm{R}}^{(k,j-1)} = \mathrm{reshape}\{\mathbf{G}_{\mathrm{R}}^{(k,j-1)}\mathrm{vec}\{\mathbf{F}^{(k)}\},[M,R]\}$, and read

$$\mathbf{\Sigma}^{(k,j)} = \mathbf{R}_{(3)}\Big(\mathbf{A}_{\mathrm{L}}^{(k,j-1)} \odot \mathbf{A}_{\mathrm{R}}^{(k,j-1)}\Big)\Big(\big[(\mathbf{A}_{\mathrm{L}}^{(k,j-1)})^{\top}\mathbf{A}_{\mathrm{L}}^{(k,j-1)}\big] \circledast \big[(\mathbf{A}_{\mathrm{R}}^{(k,j-1)})^{\top}\mathbf{A}_{\mathrm{R}}^{(k,j-1)}\big]\Big)^{\#}, \quad (24a)$$

$$\mathbf{A}_{\mathrm{L}}^{(k,j)} = \Big(\mathbf{R}_{(1)}\mathbf{Q}_{\mathrm{R}}^{(k,j-1)} + \rho_{\circ}\mathbf{T}_{\mathrm{R}}^{(k,j-1)} + \rho_{\mathbb{A}}\big(\mathbf{A}_{\mathrm{R}}^{(k,j-1)} - \mathbf{E}^{(k,j-1)}\big)\Big)\Big(\big(\mathbf{H}_{\mathrm{R}}^{(k,j-1)} + \rho_{\mathbb{A}}\mathbf{I}_R\big)^{-1}, \quad (24b)$$

$$\mathbf{A}_{\mathrm{R}}^{(k,j)} = \Big(\mathbf{R}_{(2)}\mathbf{Q}_{\mathrm{L}}^{(k,j-1)} + \rho_{\circ}\mathbf{T}_{\mathrm{L}}^{(k,j-1)} + \rho_{\mathbb{A}}\big(\mathbf{A}_{\mathrm{L}}^{(k,j-1)} - \mathbf{E}^{(k,j-1)}\big)\Big)\Big(\mathbf{H}_{\mathrm{L}}^{(k,j-1)} + \rho_{\mathbb{A}}\mathbf{I}_R\Big)^{-1}, \quad (24c)$$

$$\mathbf{E}^{(k,j)} = \mathbf{E}^{(k,j-1)} + \mathbf{A}_{\mathrm{L}}^{(k,j)} - \mathbf{A}_{\mathrm{R}}^{(k,j)}. \quad (24d)$$

Here, $\mathbf{H}_{\mathrm{L}}^{(k,j-1)} = (\mathbf{Q}_{\mathrm{L}}^{(k,j-1)})^{\top}\mathbf{Q}_{\mathrm{L}}^{(k,j-1)} + \rho_{\circ}(\mathbf{G}_{\mathrm{L}}^{(k,j-1)})^{\top}\mathbf{G}_{\mathrm{L}}^{(k,j-1)}$ and $\mathbf{H}_{\mathrm{R}}^{(k,j-1)} = (\mathbf{Q}_{\mathrm{R}}^{(k,j-1)})^{\top}\mathbf{Q}_{\mathrm{R}}^{(k,j-1)} + \rho_{\circ}(\mathbf{G}_{\mathrm{R}}^{(k,j-1)})^{\top}\mathbf{G}_{\mathrm{R}}^{(k,j-1)}$. While $\rho_{\mathbb{A}}$ and $\mathbf{E}$ play the same role as ρ_{B} and $\mathbf{D}$ in (18), respectively. Once (24) meets the stopping criteria, we estimate $\mathbf{\Pi}_{\mathbf{A}}^{(k)}$ and $\mathbf{\Lambda}_{\mathbf{A}}^{(k)}$ using a way similar to (20), and then, set $\mathbf{\Sigma}^{(k)} = \mathbf{\Sigma}^{(k,J_{\mathrm{stop}})}$ and $\mathbf{A}^{(k)} = 0.5(\mathbf{A}_{\mathrm{L}}^{(k,J_{\mathrm{stop}})} + \mathbf{A}_{\mathrm{R}}^{(k,J_{\mathrm{stop}})}\mathbf{\Pi}_{\mathbf{A}}^{(k)}\mathbf{\Lambda}_{\mathbf{A}}^{(k)})$.

3.3. Stopping Criteria, Parameter Selection & Complexity

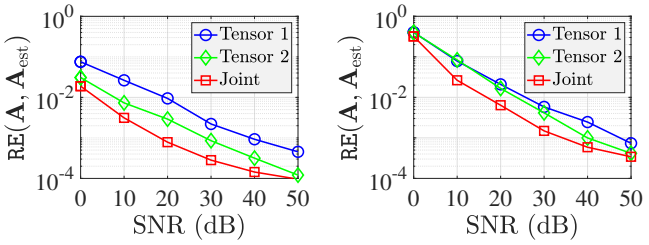
Stopping Criteria & Parameter Selection: Our method consists of an outer loop (16), and two inner loops (19) and (24). We have set their maximum number of iterations to predefined values: $K_{\mathrm{stop}} = 100$, $I_{\mathrm{stop}} = 10$, and $J_{\mathrm{stop}} = 10$, respectively. Following the guidelines in [29], we adopt the following stopping criteria, which rely on the *primal* and *dual* residuals

$$\|\mathbf{V}_{\mathrm{cur}} - \mathbf{Z}_{\mathrm{cur}}\|_F \le \varepsilon_{\mathrm{pri}}, \quad \|\rho(\mathbf{Z}_{\mathrm{cur}} - \mathbf{Z}_{\mathrm{old}})\|_F \le \varepsilon_{\mathrm{dual}}, \quad (25)$$

where "cur" and "old" represent the current and old estimates, respectively, and

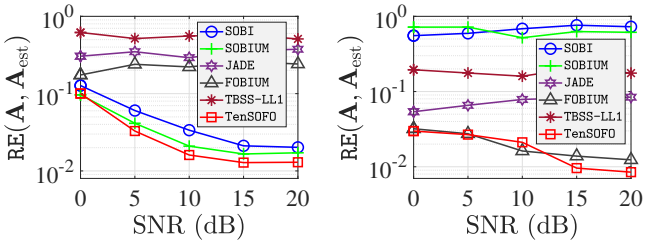
$$\varepsilon_{\mathrm{pri}} = \epsilon_{\mathrm{abs}}\sqrt{\mathrm{length}(\mathbf{V}_{\mathrm{cur}})} + \epsilon_{\mathrm{rel}}\max\big\{\|\mathbf{V}_{\mathrm{cur}}\|_2, \|\mathbf{Z}_{\mathrm{cur}}\|_2\big\},$$
$$\varepsilon_{\mathrm{dual}} = \epsilon_{\mathrm{abs}}\sqrt{\mathrm{length}(\mathbf{S}_{\mathrm{cur}})} + \epsilon_{\mathrm{rel}}\|\rho\mathbf{S}_{\mathrm{cur}}\|_2.$$

Here, $\epsilon_{\mathrm{abs}} > 0$ and $\epsilon_{\mathrm{rel}} > 0$ represent the absolute and relative tolerance, respectively. The primal variables are denoted by $\mathbf{V}$ and $\mathbf{Z}$, where $\mathbf{V}$ includes $(\mathbf{B},\mathbf{B}_{\mathrm{L}},\mathbf{A}_{\mathrm{L}})$, and $\mathbf{Z}$ includes $(\mathbf{A}_{\odot 2},\mathbf{B}_{\mathrm{R}},\mathbf{A}_{\mathrm{R}})$. The dual variables $(\mathbf{U},\mathbf{D},\mathbf{E})$ are represented by $\mathbf{S}$. While ρ represents the regularized param-



(a) $R = 3 < M$ (b) $R = 7 > M$

Fig. 1: INDSCAL decomposition: $M = 5, N = 100$, individual decomposition of $\boldsymbol{\mathcal{R}}$ (—○—), $\boldsymbol{\mathcal{C}}$ (—◇—), and joint decomposition of both $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$ (—□—).



(a) $\mathbf{s}_r = \mathbf{f}_r * \mathbf{g}_r$ (b) $\mathbf{s}_r = \text{filter}(\mathbf{v}_r, \text{AR}(1, \psi_r))$

Fig. 2: BSS task: $M = 5$, $R = 3$, $N = 30$, $T = 1000$.

eters $(\rho_{\circ}, \rho_{\text{B}}, \rho_{\text{A}})$ and their value can be selected by applying the following adaptive rule at each iteration

$$\rho^{(\ell+1)} = \begin{cases} \tau\, \rho^{(\ell)} & \text{if } \|\mathbf{V}_{\text{cur}} - \mathbf{Z}_{\text{cur}}\|_F > \mu \|\rho^{(\ell)}(\mathbf{Z}_{\text{cur}} - \mathbf{Z}_{\text{old}})\|_F \\ \rho^{(\ell)}/\tau & \text{if } \|\rho^{(\ell)}(\mathbf{Z}_{\text{cur}} - \mathbf{Z}_{\text{old}})\|_F > \mu \|\mathbf{V}_{\text{cur}} - \mathbf{Z}_{\text{cur}}\|_F \\ \rho^{(\ell)} & \text{otherwise.} \end{cases}$$

In practice, typical choices can be $\mu = 10$, $\tau = 2$, and $\rho^{(0)} = \rho_{\circ}^{(0)} = \rho_{\text{B}}^{(0)} = \rho_{\text{A}}^{(0)} = 1$, while ϵ_{abs} and ϵ_{rel} can be chosen from the ranges $[10^{-6}; 10^{-3}]$ and $[10^{-4}; 10^{-2}]$, respectively. For further details, please refer to [29].

Complexity: For short, we assume $N_1 = N_2 = N$. TenSOFO involves two inner ADMM loops, denoted as (19) and (24). In loop (19), the computation includes a pseudo-inverse and two inverse operations of $R \times R$ matrices, resulting in a cost of $\mathcal{O}(R^3)$ flops. Additionally, the Khatri-Rao products requires a cost of $\mathcal{O}(\max\{M^2, N\}M^2R)$ flops. Consequently, each iteration incurs a total cost of $\mathcal{O}(M^4R^2N)$ flops for updating $\mathbf{K}^{(k,i)}$, $\mathbf{B}_{\text{L}}^{(k,i)}$, $\mathbf{B}_{\text{R}}^{(k,i)}$, and $\mathbf{D}^{(k,i)}$. Thus, the computational complexity of (19) is $\mathcal{O}(I_{\text{stop}}M^4R^2N)$. At the end of the loop (19), TenSOFO also involves the scaling and permutation step (20) that requires $\mathcal{O}(M^4R^2)$ flops. Loop (24) shares a similar update rule with (19) but deals with $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M \times M \times N}$ of smaller size, leading to a complexity of $\mathcal{O}(J_{\text{stop}}M^2R^2N)$. To sum up, the overall computational complexity of TenSOFO is $\mathcal{O}\big(K_{\text{stop}}(I_{\text{stop}}M^2 + J_{\text{stop}})M^2R^2N\big)$ flops.

4. EXPERIMENTS

In this section, we evaluate the performance of TenSOFO in two aspects: (i) its effectiveness for joint INDSCAL decomposition, and (ii) its application to BSS using SO/FO statistics.

Experiment 1: We first apply TenSOFO to compute joint INDSCAL decomposition of two symmetric tensors $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{M \times M \times N}$ and $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{M^2 \times M^2 \times N}$ sharing the same rank R:

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}_{\text{true}} + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}} = [\![\mathbf{A}, \mathbf{A}, \boldsymbol{\Sigma}]\!] + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}, \quad \text{(Tensor 1)}$$
$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}_{\text{true}} + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}} = [\![\mathbf{A}_{\odot 2}, \mathbf{A}_{\odot 2}, \mathbf{K}]\!] + \boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}. \quad \text{(Tensor 2)}$$

Here, the tensor factors of interest $\mathbf{A} \in \mathbb{R}^{M \times R}, \boldsymbol{\Sigma} \in \mathbb{R}^{N \times R}$ and $\mathbf{K} \in \mathbb{R}^{N \times R}$ are generated as Gaussian matrices with zero-mean and unit-variance entries. $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}$ and $\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}$ represent random Gaussian noises sharing the same SNR level, i.e., $\|\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{R}}}\|_F / \|\boldsymbol{\mathcal{R}}_{\text{true}}\|_F = \|\boldsymbol{\mathcal{N}}_{\boldsymbol{\mathcal{C}}}\|_F / \|\boldsymbol{\mathcal{C}}_{\text{true}}\|_F = 10^{-\frac{\text{SNR}}{20}}$.

To evaluate the estimation accuracy, we measure the metric

$$\text{RE}(\mathbf{A}, \mathbf{A}_{\text{est}}) = \min_{\boldsymbol{\Pi}, \boldsymbol{\Lambda}} \big\|\mathbf{A} - \mathbf{A}_{\text{est}}\boldsymbol{\Pi}\boldsymbol{\Lambda}\big\|_F \big/ \big\|\mathbf{A}\big\|_F, \quad (26)$$

where $\mathbf{A}_{\text{est}}$ refers to the estimate, $\boldsymbol{\Pi}$ and $\boldsymbol{\Lambda}$ represent the permutation and scaling matrices, respectively (due to the inherent permutation and scaling indeterminacy problem). Fig. 1 illustrates the performance comparison between TenSOFO and the classical INDSCAL method (i.e., CP-ALS) for each $\boldsymbol{\mathcal{R}}$ and $\boldsymbol{\mathcal{C}}$. The results indicate that the joint decomposition significantly improves the estimation accuracy of $\mathbf{A}$ as compared to the individual decomposition approach.

Experiment 2: We then demonstrate the effectiveness of TenSOFO for BSS tasks in comparison with several widely-used BSS algorithms, namely SOBI [20], JADE [19], SOBIUM [9], FOOBI [8], and TBSS-LL1 [17].

For this task, we use the data model (1) in which the noise vector is $\mathbf{n}(t) \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}_M)$ and the mixing matrix $\mathbf{A} \in \mathbb{R}^{M \times R}$ is generated as a Gaussian matrix with zero-mean, unit-variance entries. We consider two scenarios for the source matrix $\mathbf{S} = [\mathbf{s}_1^\top, \mathbf{s}_2^\top, \ldots, \mathbf{s}_R^\top]^\top \in \mathbb{R}^{R \times T}$: (i) $\mathbf{s}_r$ is the result of convolving a kernel/filter $\mathbf{f}_r$ of length $L_f \ll T$ with a random coefficient vector $\mathbf{g}_r$ of length $T - L_f + 1$ (i.e., $\mathbf{s}_r = \mathbf{f}_r * \mathbf{g}_r$); and (ii) $\mathbf{s}_r$ is derived from filtering a non-Gaussian random process $\mathbf{v}_r$ by a first-order autoregressive (AR) model with coefficient ψ_r, denoted as $\mathbf{s}_r = \text{filter}(\mathbf{v}_r, \text{AR}(1, \psi_r))$. In the first case, $\mathbf{g}_r$ and $\mathbf{f}_r$ are generated as normal and folded-normal random vectors, respectively. Moreover, we set the filter length $L_f = 30$. In the second case, we define the non-Gaussian process for the r-th source $\mathbf{s}_r$ using a power of normal Gaussian distribution (i.e., $\mathbf{v}_t(i) = |y_i|^p$ where $y_i \sim \mathcal{N}(0, 1), p > 1$). Here, we set $p = 6$ and $\psi_r = 0.5\ \forall r$. The hyperparameters of the compared BSS algorithms are kept at their default values. SOBI, SOBIUM, and TenSOFO require a predefined number of time lags, we set its value to $N = 30$. To evaluate the estimation accuracy of the BSS algorithms, we reuse the error metric $\text{RE}(\mathbf{A}, \mathbf{A}_{\text{est}})$ in (26). Fig. 2 indicates that TenSOFO outperforms other algorithms, offering better estimation accuracy in both test cases.

5. CONCLUSIONS

In this paper, we addressed the problem of blind source separation (BSS) using both second and fourth order statistics. We established a fundamental connection between joint INDSCAL decomposition and BSS, which served as the basis for introducing our novel tensor-based method called TenSOFO. The proposed method is specifically designed for joint INDSCAL decomposition and, consequently, BSS tasks. The experimental results indicated the effectiveness of TenSOFO, showcasing its excellent performance in both tensor decomposition and blind source separation tasks, particularly when compared to state-of-the-art algorithms.

6. REFERENCES

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 2010.

[2] G. R. Naik and W. Wang, *Blind Source Separation: Advances in Theory, Algorithms and Applications*, 2014.

[3] G. Chabriel, M. Kleinsteuber, E. Moreau *et al.*, "Joint matrices decompositions and blind source separation: A survey of methods, identification, and applications," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 34–43, 2014.

[4] L. De Lathauwer *et al.*, "Blind source separation by higher-order singular value decomposition," in *Proc. Eur. Signal Process. Conf.*, 1994, pp. 175–178.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Independent component analysis and (simultaneous) third-order tensor diagonalization," *IEEE Trans. Signal Process.*, vol. 49, no. 10, pp. 2262–2271, 2001.

[6] 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.

[7] C. F. Beckmann and S. M. Smith, "Tensorial extensions of independent component analysis for multisubject fMRI analysis," *Neuroimage*, vol. 25, no. 1, pp. 294–311, 2005.

[8] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2965–2973, 2007.

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

[10] M. Sørensen and L. De Lathauwer, "Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5507–5519, 2013.

[11] D. Nion, K. N. Mokios, N. D. Sidiropoulos, and A. Potamianos, "Batch and adaptive PARAFAC-based blind separation of convolutive speech mixtures," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 6, pp. 1193–1207, 2010.

[12] X.-F. Gong, Q.-H. Lin, F.-Y. Cong, and L. De Lathauwer, "Double coupled canonical polyadic decomposition for joint blind source separation," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3475–3490, 2018.

[13] L. De Lathauwer, "Block component analysis, a new concept for blind source separation," in *Int. Conf. Latent Var. Anal. Signal Sep.*, 2012, pp. 1–8.

[14] N. Govindarajan, E. N. Epperly, and L. De Lathauwer, "(L_r,L_r,1)-decompositions, sparse component analysis, and the blind separation of sums of exponentials," *SIAM J. Matrix Anal. Appl.*, vol. 43, no. 2, pp. 912–938, 2022.

[15] L. De Lathauwer, "Blind separation of exponential polynomials and the decomposition of a tensor in rank-(L_r,L_r,1) terms," *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 4, pp. 1451–1474, 2011.

[16] O. Debals, M. Van Barel, and L. De Lathauwer, "Löwner-based blind signal separation of rational functions with applications," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 1909–1918, 2016.

[17] M. Boussé, O. Debals, and L. De Lathauwer, "A tensor-based method for large-scale blind source separation using segmentation," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 346–358, 2017.

[18] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proc. IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

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

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

[21] A. Ferreol and P. Chevalier, "On the behavior of current second and higher order blind source separation methods for cyclostationary sources," *IEEE Trans. Signal Process.*, vol. 48, no. 6, pp. 1712–1725, 2000.

[22] K. Abed-Meraim, Y. Xiang, J. H. Manton, and Y. Hua, "Blind source-separation using second-order cyclostationary statistics," *IEEE Trans. Signal Process.*, vol. 49, no. 4, pp. 694–701, 2001.

[23] A. Ferreol, L. Albera, and P. Chevalier, "Fourth-order blind identification of underdetermined mixtures of sources (FOBIUM)," *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1640–1653, 2005.

[24] L. Albera, A. Ferréol, P. Chevalier, and P. Comon, "ICAR: a tool for blind source separation using fourth-order statistics only," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3633–3643, 2005.

[25] A. Aissa-El-Bey, K. Abed-Meraim, Y. Grenier, and Y. Hua, "A general framework for second-order blind separation of stationary colored sources," *Signal Process.*, vol. 88, no. 9, pp. 2123–2137, 2008.

[26] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[27] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[28] I. Domanov and L. De Lathauwer, "Generic uniqueness conditions for the canonical polyadic decomposition and INDSCAL," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1567–1589, 2015.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.