Multilinear Generalized Singular Value Decomposition (ML-GSVD) and Its Application to Multiuser MIMO Systems

Liana Khamidullina, *Student Member, IEEE*, André L. F. de Almeida, *Senior Member, IEEE*, and Martin Haardt, *Fellow, IEEE*

***Abstract*—In this paper, we introduce a Multilinear Generalized Singular Value Decomposition (ML-GSVD) for two or more matrices with one common dimension. The ML-GSVD extends the Generalized Singular Value decomposition (GSVD) of two matrices to higher orders. The proposed decomposition allows us to jointly factorize a set of matrices with one common dimension. In comparison with other approaches that extend the GSVD, the ML-GSVD preserves the essential properties of the original (matrix-based) GSVD, such as orthogonality of the second-mode factor matrices as well as the subspace structure of the third-mode factor matrices. We introduce an ALS-based algorithm to compute the ML-GSVD, which has been inspired by PARAFAC2 decomposition algorithms. In addition, we present an application of the ML-GSVD for transceiver optimization in multicast and unicast MIMO-OFDM systems. Our numerical results show that the proposed ML-GSVD multicast and unicast beamforming outperforms existing state-of-the-art schemes in terms of the sum rate.**

***Index Terms*—Tensor decomposition, GSVD, ML-GSVD, MIMO, coordinated beamforming, broadcasting.**

I. Introduction

DURING the last decades, interest in tensor-based signal processing methods has exponentially grown due to their advantages over the conventional matrix-based methods. The higher-order extensions of the matrix decompositions enable their generalization to multiway data processing. Tensor techniques allow us to exploit the original structure of the multidimensional data in many applications.

Manuscript received December 21, 2021; revised April 29, 2022; accepted May 18, 2022. Date of publication May 30, 2022; date of current version June 14, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Nicolas Gillis. The work of André L. F. de Almeida was supported by CNPq under Grant 312491/2020-4. This work was supported in part by German Research Foundation under Grants HA 2239/14-1 (AdAMMM) and ZH 640/2-1 (AdAMMM), in part by CAPES/PROBRAL Proc. under Grant 88887.144009/2017-00, and in part by CAPES/PRINT Proc. under Grant 88887.311965/2018-00. *(Corresponding author: Liana Khamidullina.)*

Liana Khamidullina and Martin Haardt are with Communication Research Laboratory, Ilmenau University of Technology, 98693 Ilmenau, Germany (e-mail: liana.khamidullina@tu-ilmenau.de; martin.haardt@tu-ilmenau.de).

André L. F. de Almeida is with Wireless Telecommunications Research Group, Federal University of Ceará, Fortaleza 60020-181, Brazil (e-mail: andre@gtel.ufc.br).

Digital Object Identifier 10.1109/TSP.2022.3178902

A lot of existing tensor decompositions generalize the matrix decompositions. Despite the fact that there already exist multidimensional extensions of the Generalized Singular Value Decomposition (GSVD) in the literature, none of them fully inherits the features of the original decomposition. On the contrary, the Multilinear Generalized Singular Value Decomposition (ML-GSVD) that we have first presented in [1] simplifies to the GSVD when performed on two matrices and extends its properties to higher orders. The GSVD is useful in various communication and biomedical applications, such as coordinated beamforming [2]–[4], MIMO relaying [5], [6], physical layer security [7], [8], and genomic signal processing [9]–[11]. In this paper, we present an extension of the GSVD [12], [13] to factorize a three-way tensor. The proposed ML-GSVD can be used for the joint analysis of a collection of an arbitrary number (more than two) matrices with a (possibly) varying number of rows and the same number of columns. Thus, the ML-GSVD has the potential to be employed in the current GSVD applications without being limited to two matrices. We have already presented some initial investigations on the ML-GSVD and its calculation in [1]. In this contribution, we introduce a more general definition of the ML-GSVD which is applicable to a set of both full-rank and rank-deficient matrices. Moreover, we specify the corresponding optimization problem. Additionally, the paper presents an enhanced algorithm to calculate the ML-GSVD and its application to multiuser MIMO-OFDM systems with multicast and unicast transmissions.

Before presenting the main differences and advantages of the ML-GSVD over other GSVD extensions, let us first give a brief overview of the existing generalizations of the GSVD. Two different multidimensional decompositions that extend the GSVD to the tensor case have been introduced in [14] and [15]. The authors in [14] define the Higher Order Generalized Singular Value Decomposition (HO GSVD) as an exact decomposition of two or more full-rank real-valued matrices that preserves some properties of the matrix-based GSVD. However, it does not preserve the orthogonality of the factor matrices as in the matrix GSVD. The HO GSVD in [14] is used for a comparative analysis of the global mRNA expression datasets from different organisms. The "common HO GSVD subspace" represents the similarity in three different organisms. However, the definition of the "common HO GSVD subspace" does not exactly match

the representation of the common and private subspaces in the original GSVD. In contrast to the aforementioned paper, the authors in [16] do not restrict their decomposition to full-rank matrices and present additional steps that enable the application of the HO GSVD to rank deficient third-mode slices. In [15], the authors have presented a Tensor GSVD to jointly decompose two tensors with the matched column but independent row dimensions. The decomposition is then used for DNA analysis. Both [14] and [15] consider real-valued matrices in a biomedical data processing context. In contrast to the HO GSVD and the Tensor GSVD, the ML-GSVD proposed in our work, inherits most of the properties of the original GSVD and is applicable to both real and complex-valued data, which paves the way for its use in many signal processing applications, such as in wireless communications, where the data tensors usually represent the communication channel and/or the received signal in their equivalent complex-valued baseband form.

Due to the fact that the ML-GSVD provides orthogonal factor matrices for the individual slices, it is a valuable tool for coordinated downlink beamforming in a wireless multiuser MIMO system. More specifically, by applying the ML-GSVD to a set of channel matrices (associated with different users), we are able to identify common subspaces (CSs) to a group of users, as well as private subspaces (PSs) to individual users. Hence, by exploiting the structure of these subspaces, broadcast and multicast transmission can be simultaneously combined on the downlink for any number of users. In [5], [6], and [7] the SVD-based beamforming has been generalized to GSVD-based MIMO downlink beamforming. The authors illustrate how the GSVD can be exploited for coordinated beamforming in a multiuser MIMO system, but the approach is limited to two users. The authors in [4] propose the GSVD for polynomial matrices (PGSVD) and present its application to two-user frequency-selective MIMO channels. In [17], [18] the GSVD is combined with a non-orthogonal multiple access (NOMA) scheme in a MIMO downlink scenario. The use of the ML-GSVD allows us to go further by increasing the number of users to be simultaneously served. In [1], we show how the ML-GSVD can be used for coordinated beamforming in a multiuser MIMO downlink channel with more than two users. Moreover, in [19] we apply the ML-GSVD to a NOMA communication system with an arbitrary number of users. Depending on the number of transmit and receive antennas (tensor dimensions), the subspace structure of the ML-GSVD distinguishes between common and private subspaces. Common subspaces are used to transmit the same data to several users, while private subspaces allow sending confidential messages to different users simultaneously. Hence, the ML-GSVD enables handling an arbitrary number of users that is less or equal than the number of transmit antennas in the downlink of a coordinated MIMO beamforming system.

The authors in [20] consider a MIMO-OFDM multicasting system and propose an SVD-based non-iterative linear precoding scheme that allows sending common messages to a group of users. Various multicast precoding methods have also been studied in [21]–[30]. The authors in [31] use zero forcing (ZF), minimum mean square error (MMSE), and signal-to-interference-plus-noise ratio (SINR) balancing for the multicast transmission. In contrast to [20], the approach of [31] assumes a multigroup multicast scenario and a single-carrier system with single antenna users.

Most of the proposed techniques in the literature are limited to pure multicast or unicast transmission, but practical systems are interested in joint services. This has motivated the study of a joint unicast and multicast transmission. However, the existing techniques have so far been limited to a simple system with two users or multiple single-antenna users. None of the aforementioned papers considers a joint multicast and unicast precoding for a MIMO-OFDM system with multiple antenna users. In our paper, we show that the ML-GSVD is a valuable tool that allows combining broadcast and unicast transmissions, which leads to an increased average spectral efficiency. The main contributions of this paper can be summarized as follows:

- We define a new extension of the GSVD [12], [13] to factorize a three-way tensor. The proposed multilinear generalized singular value decomposition (ML-GSVD) can be used for the joint analysis of a collection of more than two matrices with a varying number of rows and the same number of columns. In contrast to other GSVD extensions, the ML-GSVD inherits the properties of the original matrix-based GSVD. Moreover, we do not place any restrictions on the matrix ranks and consider three different cases depending on the dimensions of the decomposed matrices. We also show that in one of these cases the decomposition is exact.
- A general ALS-based algorithm is introduced to compute the proposed ML-GSVD decomposition. In particular, we show that any algorithm to calculate the PARAFAC2 decomposition can be modified to compute the proposed ML-GSVD.
- Since the ML-GSVD of two matrices with one common dimension is exact and equal to the GSVD, the proposed algorithm leads to an alternative way to calculate the GSVD of two matrices.
- As one of the promising applications of the ML-GSVD, we consider a MIMO-OFDM system with joint unicast and multicast transmissions. We show that the factor matrices of the ML-GSVD can be used as the precoding and decoding matrices, respectively. We also demonstrate that the aforementioned three cases of the decomposition correspond to the transmission of private or common messages (or both). The simulation results show that the ML-GSVD outperforms the reference multicast and unicast precoding schemes in terms of the sum rate.

This paper is organized as follows. In Section II, we review the GSVD for two matrices. Then, we introduce the ML-GSVD in Section III. Section IV presents an algorithm to compute the ML-GSVD. The numerical results are shown in Section V. In Section VI, we present an application of the ML-GSVD to coordinated beamforming in multiuser MIMO-OFDM systems. The paper is concluded in Section VII.

Notation: Matrices and vectors are denoted by upper-case and lower-case bold-faced letters, respectively. Bold-faced calligraphic letters denote tensors. The superscripts $\{\cdot\}^{\mathrm{T}}$ and $\{\cdot\}^{\mathrm{H}}$ denote the transpose and Hermitian transpose, respectively,

whereas $\mathrm{diag}\{\cdot\}$ is the operation of constructing a diagonal matrix with diagonal elements being the entries of the input vector, while $\mathrm{bdiag}\{\cdot\}$ is the operation of constructing a block diagonal matrix with the input matrices on the main diagonal. The j-th row and the i-th column of a matrix $\boldsymbol{A} \in \mathbb{C}^{J\times I}$ is represented by $\boldsymbol{A}(j,:) \in \mathbb{C}^{I}$ and $\boldsymbol{A}(:,i) \in \mathbb{C}^{J}$, respectively, where $i = 1,\ldots,I$ and $j = 1,\ldots,J$. The Kronecker and Khatri-Rao products are denoted as $\otimes$ and $\diamond$, respectively. Additionally, we denote the higher-order norm of a tensor $\boldsymbol{\mathcal{A}}$ by $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{H}}$, the two-norm of a vector $\boldsymbol{a}$ by $\|\boldsymbol{a}\|$, and $\|\boldsymbol{A}\|_{\mathrm{F}}$ denotes the Frobenius norm. The r-mode unfolding of the tensor $\boldsymbol{\mathcal{A}}$ is denoted as $[\boldsymbol{\mathcal{A}}]_{(r)}$ (we use the reverse cyclical ordering of the columns [32]). $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix. $\boldsymbol{0}_x$ denotes the vector of zeros of size x and $\boldsymbol{O}_x$ denotes the $x \times x$ matrix of zeros. $\{\boldsymbol{A}\}_{\mathcal{R}(\mathcal{Q})}$ and $\{\boldsymbol{A}\}_{\mathcal{C}(\mathcal{Q})}$ denote the sub-matrices consisting of the columns and rows of $\boldsymbol{A}$ with indices in the set $\mathcal{Q} \subseteq \{1,\ldots,I\}$, respectively.

II. Generalized Singular Value Decomposition

Before introducing the ML-GSVD, let us first review the GSVD of two matrices proposed in [12], [13], and [33]. Let $\boldsymbol{H}_1 \in \mathbb{C}^{J_1\times I}$ and $\boldsymbol{H}_2 \in \mathbb{C}^{J_2\times I}$ be two matrices having the same number of columns, and an arbitrary number of rows J_1 and J_2. Then, the GSVD of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ is defined as

$$\begin{aligned}\boldsymbol{H}_1 &= \boldsymbol{B}_1 \cdot \boldsymbol{C}_1 \cdot \boldsymbol{A}^{\mathrm{H}},\\ \boldsymbol{H}_2 &= \boldsymbol{B}_2 \cdot \boldsymbol{C}_2 \cdot \boldsymbol{A}^{\mathrm{H}},\end{aligned} \tag{1}$$

where $\boldsymbol{B}_1 \in \mathbb{C}^{J_1\times J_1}$ and $\boldsymbol{B}_2 \in \mathbb{C}^{J_2\times J_2}$ have orthogonal columns, $\boldsymbol{A} \in \mathbb{C}^{I\times I}$ is nonsingular and common for both matrices. Moreover, $\boldsymbol{C}_1 \in \mathbb{R}^{J_1\times I}$ and $\boldsymbol{C}_2 \in \mathbb{R}^{J_2\times I}$ are non-negative diagonal matrices. The ratios of the corresponding entries of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are called generalized singular values of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. Let $q = \mathrm{rank}([\boldsymbol{H}_1^{\mathrm{H}} \;\; \boldsymbol{H}_2^{\mathrm{H}}])$, $r = q - \mathrm{rank}(\boldsymbol{H}_2)$, and $s = \mathrm{rank}(\boldsymbol{H}_1) + \mathrm{rank}(\boldsymbol{H}_2) - q$, then, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ have the following structure

$$\boldsymbol{C}_1 = \begin{bmatrix} \boldsymbol{O}_{(J_1-r-s)\times(q-r-s)} & & \\ & \hat{\boldsymbol{\Sigma}} & \\ & & \boldsymbol{I}_r \end{bmatrix}, \tag{2}$$

$$\boldsymbol{C}_2 = \begin{bmatrix} \boldsymbol{I}_{(q-r-s)} & & \\ & \hat{\boldsymbol{\Lambda}} & \\ & & \boldsymbol{O}_{(J_2-q+r)\times(r)} \end{bmatrix}, \tag{3}$$

where $\boldsymbol{I}_r$ and $\boldsymbol{I}_{(q-r-s)}$ are identity matrices, $\boldsymbol{O}_{(J_1-r-s)\times(q-r-s)}$ and $\boldsymbol{O}_{(J_2-q+r)\times(r)}$ are zero matrices possibly having no rows or no columns, $\hat{\boldsymbol{\Sigma}} = \mathrm{diag}(\sigma_1,\ldots,\sigma_s)$, $\hat{\boldsymbol{\Lambda}} = \mathrm{diag}(\lambda_1,\ldots,\lambda_s)$ are diagonal matrices, such that $0 < \sigma_n < 1, 0 < \lambda_n < 1$, and $\sigma_n^2 + \lambda_n^2 = 1$ for $n \in \{1,\ldots,s\}$. There are some small variations in the representation of the GSVD, for instance, in [13] it is assumed that $J_2 \geq I$. However, most of the algorithms to calculate the GSVD commonly consist of a sequential computation of the QR and cosine-sine (CSD) decompositions. For more details on the GSVD and its applications, we refer the reader to [12], [13], and [34].

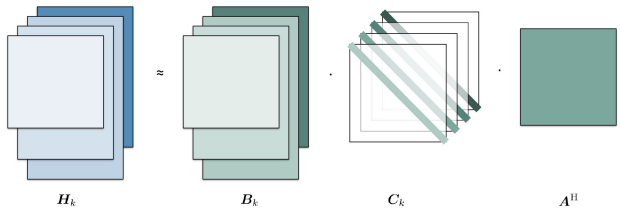


Fig. 1. Illustration of Multilinear Generalized Singular Value Decomposition.

III. Multilinear Generalized Singular Value Decomposition (ML-GSVD)

We define the ML-GSVD for a set of $K \geq 2$ complex valued matrices[1] $\boldsymbol{H}_k \in \mathbb{C}^{J_k\times I}$ with the same column dimension and possibly different row dimensions as follows

$$\begin{aligned}\boldsymbol{H}_1 &= \boldsymbol{B}_1 \cdot \boldsymbol{C}_1 \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_1\times I},\\ &\vdots\\ \boldsymbol{H}_K &= \boldsymbol{B}_K \cdot \boldsymbol{C}_K \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_K\times I}.\end{aligned} \tag{4}$$

The K matrices can be viewed as 3-mode slices of the tensor $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{J\times I\times K}$, where $J = \max(J_1,\ldots,J_K)$ (zeros are added for those elements that are not defined in (4)). The tensor representation enables the use of tensor-based algorithms to compute the matrices $\boldsymbol{A}, \boldsymbol{C}_k$, and $\boldsymbol{B}_k$. Subsequently, the ML-GSVD of the tensor $\boldsymbol{\mathcal{H}}$ (Fig. 1) can be defined in a slice-wise fashion as

$$\boldsymbol{H}_k = \boldsymbol{B}_k \cdot \boldsymbol{C}_k \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_k\times I}, \tag{5}$$

where $k \in \{1,\ldots,K\}$ is the kth slice of $\boldsymbol{\mathcal{H}}$. The matrix $\boldsymbol{A} \in \mathbb{C}^{I\times Q}$, $Q = \min\{\sum_{k=1}^{K} J_k, I\}$, is nonsingular and common for all the K 3-mode slices of the tensor $\boldsymbol{\mathcal{H}}$. The matrix $\boldsymbol{B}_k \in \mathbb{C}^{J_k\times Q}$ corresponding to the k-th slice of $\boldsymbol{\mathcal{H}}$ has orthogonal columns such that $\boldsymbol{B}_k^{\mathrm{H}} \cdot \boldsymbol{B}_k = \boldsymbol{I}_Q$. The matrices $\boldsymbol{C}_k \in \mathbb{R}^{Q\times Q}$ are diagonal with non-negative entries satisfying $\sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q$. We additionally define a matrix $\boldsymbol{C} \in \mathbb{R}^{K\times Q}$ in which the diagonal elements of $\boldsymbol{C}_k$ are stacked as rows, i.e., $\boldsymbol{C}_k = \mathrm{diag}\{\boldsymbol{C}(k,:)\}$. Then, $\boldsymbol{C}$ has unit column norms: $\sum_{k=1}^{K} c_{k,i}^2 = 1$ for all $i = 1 \ldots Q$. Consequently, the optimization problem to be solved can be stated as follows

$$\begin{aligned}\underset{\boldsymbol{A},\boldsymbol{B}_k,\boldsymbol{C}_k}{\text{minimize}} \quad & \sum_{k=1}^{K} \|\boldsymbol{H}_k - \boldsymbol{B}_k\boldsymbol{C}_k\boldsymbol{A}^{\mathrm{H}}\|_{\mathrm{F}}^2\\ \text{s.t.} \quad & \boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{I}_Q, \;\; \sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q.\end{aligned} \tag{6}$$

It should be noted that due to the imposed constraints, the model in (5) is an approximation of $\boldsymbol{H}_k$ in a least squares sense. However, in the following we will show that in some cases the decomposition is exact.

As for the computation of the GSVD in [33], we assume that the null spaces of the $\boldsymbol{H}_k$s do not overlap, i.e., $\mathrm{null}(\boldsymbol{H}_1) \cap$

[1] In general, the proposed decomposition is also applicable to real-valued matrices. Since we further focus on an application in communications, we consider the decomposition of complex-valued matrices in this paper.

$\text{null}(\boldsymbol{H}_2) \cap \cdots \cap \text{null}(\boldsymbol{H}_K) = \emptyset$. The elements of $\boldsymbol{C} \in \mathbb{R}^{K \times Q}$ are non-negative, and the columns of $\boldsymbol{C}$ have unit norm. The values of the first row of $\boldsymbol{C}$ are sorted in ascending order, such that the first row of $\boldsymbol{C}$ has the following structure

$$\boldsymbol{C}(1,:) = \begin{bmatrix} \mathbf{0}_{p_2+\cdots+p_K}^{\mathrm{T}} & \boldsymbol{\sigma}_1^{\mathrm{T}} & \mathbf{1}_{p_1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{1 \times Q}, \tag{7}$$

where $\boldsymbol{C}(1,:) = \text{diag}\{\boldsymbol{C}_1\}$, $\mathbf{1}_{p_1}$ is a vector of ones, and $\mathbf{0}_{p_2+\cdots+p_K}$ is a vector of zeros, which might have no entries. The values of $\boldsymbol{\sigma}_k \in \mathbb{R}^{c_k}$ are in the range (0,1), c_k and p_k are the dimensions of the *common* and *private* subspaces, respectively. The remaining rows of $\boldsymbol{C}$ are sorted according to the first row. Whenever there are ambiguities (elements that correspond to zeros in the first row), we sort the elements of the second row in ascending order. Then, we turn to the third row, the elements of which are sorted according to the first and second rows. Whenever there are ambiguities (zeros in the previous row), we sort the elements of the third row in ascending order. After that, we switch to the fourth row, and so on. Then, the columns of $\boldsymbol{A}$ and $\boldsymbol{B}_k$ are rearranged accordingly. Generally, a permutation of the elements in $\boldsymbol{C}$ does not change the meaning of the decomposition and is performed for notational simplicity. The vectors of ones represent the private subspace of the kth matrix, whereas $\boldsymbol{\sigma}_k$ corresponds to the common subspace of all matrices or a group of matrices. Since $\sum_{k=1}^{K} c_{k,i}^2 = 1$, the private subspace of the matrix $\boldsymbol{H}_k$ always coincides with the zero vectors of the remaining matrices. The numerical example in (8) and (9) below ($I = 6, J_1 = J_2 = J_3 = J_4 = 2$) illustrates the matrix $\boldsymbol{C}$ before and after the reordering of the columns. First, the columns are permuted in ascending order according to the first row. Then, the columns one to four (that correspond to zero elements in the first row) are reordered according to the ascending order of the second row. Next, the rearrangement is applied to the columns one, two, and three that coincide with zeros in the second row.

$$\boldsymbol{C}_{\text{before reordering}} = \begin{bmatrix} 1.0000 & 0.7050 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.7092 & 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.9193 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.3935 & 0.0000 & 1.0000 \end{bmatrix}, \tag{8}$$

$$\boldsymbol{C}_{\text{after reordering}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.7050 & 1.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.7092 & 0.0000 \\ 0.0000 & 0.9193 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.0000 & 0.3935 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{bmatrix}. \tag{9}$$

Depending on the dimensions and the individual ranks of the $\boldsymbol{H}_k$s, where $r_k = \text{rank}\{\boldsymbol{H}_k\}$, we distinguish three different cases:

Case 1: $r_k = I$ for $\forall k$. In this case, the matrix $\boldsymbol{C}$ has the following structure:

$$\boldsymbol{C}(k,:) = \begin{bmatrix} \boldsymbol{\sigma}_k^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{1 \times I}, \tag{10}$$

where $\boldsymbol{\sigma}_k^{\mathrm{T}} = [\sigma_{k,1}, \ldots, \sigma_{k,I}]$, such that $1 > \sigma_{k,i} > 0$ for $i \in \{1, \ldots, I\}$. The decomposition has the following form

$$\boldsymbol{H}_k = \boldsymbol{B}_k \cdot \underbrace{\begin{bmatrix} \sigma_{k,1} & & \\ & \ddots & \\ & & \sigma_{k,I} \end{bmatrix}}_{\text{diag}\{\boldsymbol{C}(k,:)\}} \cdot \boldsymbol{A}^{\mathrm{H}} \in \mathbb{C}^{J_k \times I}, \tag{11}$$

where $\text{diag}\{\boldsymbol{C}(k,:)\}$ is a full-rank diagonal matrix. The columns of $\boldsymbol{A}$ are shared for all factorizations, and the decomposition provides only the *common* subspace of size I for all the matrix slices $\boldsymbol{H}_k$, $k = 1, \ldots, K$. If the matrices $\boldsymbol{H}_k$ have full rank, then the decomposition has the form as in (11) if $I \leq J_K$, otherwise Case 2 applies.

Case 2: In this case, $r_k < I$ for some k, with $\sum_{k=1}^{K} r_k > I$. This configuration provides both private and common subspaces. The dimensions of these subspaces depend on the realization of the tensor and the sizes of the matrices $\boldsymbol{H}_k$. The zeros and ones correspond to the private subspace, while the other non-zero terms correspond to the common subspace. Such a configuration implies a common subspace for all the K slices $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_K$, or for some groups of matrices. In this case, in general, the decomposition has the following structure

$$\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{O}_{J_k \times I - r_k} & \hat{\boldsymbol{B}}_k \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{O}_{I-r_k} & & \\ & \boldsymbol{\Sigma}_{c_k} & \\ & & \boldsymbol{I}_{p_k} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{A}_{o_k}^{\mathrm{H}} \\ \boldsymbol{A}_{c_k}^{\mathrm{H}} \\ \boldsymbol{A}_{p_k}^{\mathrm{H}} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{\Sigma}_{c_k} = \text{diag}\{\boldsymbol{\sigma}_k\}$ and $\hat{\boldsymbol{B}}_k \in \mathbb{C}^{J_k \times r_k}$. The matrix $\boldsymbol{A}$ is partitioned into submatrices $\boldsymbol{A}_{o_k}, \boldsymbol{A}_{c_k}$, and $\boldsymbol{A}_{p_k}$ of size $I \times (I - r_k)$, $I \times c_k$, and $I \times p_k$, respectively. The values of c_k and p_k are defined by the realization and the dimensions of the $\boldsymbol{H}_k$s, and $c_k + p_k = r_k$. The submatrix $\boldsymbol{A}_{c_k}$ is associated with the common subspace of the kth and some (or all) other matrices, and the submatrix $\boldsymbol{A}_{p_k}$ corresponds to the private subspace of $\boldsymbol{H}_k$. The matrix $\boldsymbol{A}_{o_k}$ is associated with the private and common subspaces of other matrices than the matrix k. While the matrix $\boldsymbol{A}_{c_k}$ is shared between the corresponding matrices, the matrices $\boldsymbol{A}_{p_k}$ are specific for each factorization and insure a private subspace. Note that depending on the dimensions of the different 3-mode slices $\boldsymbol{H}_k$, $k \in \{1, \ldots, K\}$, the number of common and private subspaces is not the same for all the K slices. This means that for a set of values of k, the decomposition provides both common and private subspaces, while for the remaining set, only common subspaces exist. For full rank matrices satisfying $I \geq J_k$ for some or all slices k, the decomposition leads to (12). For $K = 2$, the ML-GSVD of two matrices corresponds to the matrix-based GSVD. The dimensionality of the common subspace for $K = 2$ is $r_1 + r_2 - I$ and the dimensionality of the two private subspaces are $I - r_2$ and $I - r_1$, respectively.

Case 3: $\sum_{k=1}^{K} r_k \leq I$. In this case, the rows of $\boldsymbol{C}$ contain only ones and zeros, and their ordering is defined as follows

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{0}_{p_2}^{\mathrm{T}} & \mathbf{1}_{p_1}^{\mathrm{T}} \\ \mathbf{0}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{1}_{p_2}^{\mathrm{T}} & \mathbf{0}_{p_1}^{\mathrm{T}} \\ & \vdots & & \\ \mathbf{1}_{p_K}^{\mathrm{T}} & \cdots & \mathbf{0}_{p_2}^{\mathrm{T}} & \mathbf{0}_{p_1}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{K \times Q}, \tag{13}$$

where $Q = \min\{\sum_{k=1}^{K} J_k, I\}$. The dimensions of the private subspaces ($\mathbf{1}_{p_k}$) are equal to r_k. As it can be seen from (13), this case provides only private subspaces for each matrix $\boldsymbol{H}_k$, and therefore, the common factor matrix can be rewritten as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_K & \cdots & \boldsymbol{A}_1 \end{bmatrix}. \tag{14}$$

Then, we obtain:

$$\begin{aligned} \boldsymbol{H}_1 &= \begin{bmatrix} \boldsymbol{O}_{J_1 \times Q - p_1} & \hat{\boldsymbol{B}}_1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \boldsymbol{O}_{Q-p_1} & \\ & \boldsymbol{I}_{p_1} \end{bmatrix}}_{\boldsymbol{C}_k} \cdot \begin{bmatrix} \boldsymbol{A}_K^{\mathrm{H}} \\ \vdots \\ \boldsymbol{A}_1^{\mathrm{H}} \end{bmatrix} \\ &= \hat{\boldsymbol{B}}_1 \boldsymbol{A}_1^{\mathrm{H}} \\ \boldsymbol{H}_2 &= \hat{\boldsymbol{B}}_2 \boldsymbol{A}_2^{\mathrm{H}} \\ &\vdots \\ \boldsymbol{H}_K &= \hat{\boldsymbol{B}}_K \boldsymbol{A}_K^{\mathrm{H}} \end{aligned} \tag{15}$$

As it can be observed from (15), the matrix $\boldsymbol{C}_k$ can be viewed as a selection matrix, which separates $\boldsymbol{A}$ into the subblocks that are not shared between the different $\boldsymbol{H}_k$s. Therefore, the matrices in this case do not have any common factors. All $\boldsymbol{H}_k$s are decomposed separately, and the submatrices in $\boldsymbol{A}$ correspond to a private subspace of each matrix. Equation (15) shows that by multiplying $\hat{\boldsymbol{B}}_1$ from the right hand side by an arbitrary non-singular matrix $\boldsymbol{T}$ and premultiplying $\boldsymbol{A}_1^{\mathrm{H}}$ by $\boldsymbol{T}^{-1}$ gives the same function value. The decomposition is essentially unique when it is subject only to this indeterminacy. For the full-rank $\boldsymbol{H}_k$s, i.e., $\mathrm{rank}\{\boldsymbol{H}_k\} = J_k$, the decomposition has the form as in (13) if $\sum_{k=1}^{K} J_k \leq I$.

IV. Computation of the ML-GSVD Inspired by PARAFAC2 Algorithms

From equation (5), it can be observed that the ML-GSVD has the same representation as the PARAFAC2 decomposition [35]. It is known from [35] that the uniqueness of the PARAFAC2 decomposition (up to column permutation and scaling) is ensured by the Harshman constraint $\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{F}^{\mathrm{H}}\boldsymbol{F}$, such that $\boldsymbol{B}_k^{\mathrm{H}} = \boldsymbol{F}^{\mathrm{T}}\boldsymbol{V}_k, \boldsymbol{V}_k\boldsymbol{V}_k^{\mathrm{H}} = \boldsymbol{I}_R$. In the ML-GSVD, since the matrices $\boldsymbol{B}_k$ are orthogonal, we set $\boldsymbol{F}$ to the identity matrix, which implies $\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{I}_Q$. By extending the GSVD to the tensor case, we also impose additional nonnegativity and unit norm constraints to the diagonal matrices $\boldsymbol{C}_k$. The similarity between PARAFAC2 and the ML-GSVD motivates us to extend efficient algorithms for PARAFAC2 to compute the ML-GSVD, as discussed in this section.

The ML-GSVD optimization problem can be formulated as follows

$$\begin{aligned} \underset{\boldsymbol{A},\boldsymbol{B}_k,\boldsymbol{C}_k}{\text{minimize}} \quad & \sum_{k=1}^{K} \|\boldsymbol{H}_k - \boldsymbol{B}_k\boldsymbol{C}_k\boldsymbol{A}^{\mathrm{H}}\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{I}_Q, \quad \sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q, \end{aligned} \tag{16}$$

where $\boldsymbol{C}_k = \mathrm{diag}\{\boldsymbol{C}(k,:)\}$ is a diagonal matrix with non-negative entries, and the matrix $\boldsymbol{C} \in \mathbb{R}^{K \times Q}$ has unit norm columns: $\sum_{k=1}^{K} c_{k,i}^2 = 1$ for all $i = \{1, \ldots, Q\}$.

To compute the ML-GSVD, we propose an algorithm that has been inspired by the Direct Fitting algorithm for PARAFAC2 in [36]. To this end, we alternately minimize (16) over $\boldsymbol{B}_k$ for fixed $\boldsymbol{A}$ and $\boldsymbol{C}$, and over $\boldsymbol{A}$ and $\boldsymbol{C}$ for a fixed $\boldsymbol{B}_k$. The main steps of the algorithm are summarized in Algorithm 1.

In the first step, the algorithm is initialized with the values of $\boldsymbol{A}$ based on the left singular vectors of $\sum_{k=1}^{K} \boldsymbol{H}_k\boldsymbol{H}_k^{\mathrm{H}}$ (SVD-based initialization) and with a random non-negative matrix $\boldsymbol{C}$ satisfying $\sum_{k=1}^{K} \boldsymbol{C}_k^2 = \boldsymbol{I}_Q$. The unitary matrix $\boldsymbol{B}_k$ is updated in the second step via minimizing

$$\sum_{k=1}^{K} \|\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{H}_k - \boldsymbol{C}_k\boldsymbol{A}^{\mathrm{H}}\|_{\mathrm{F}}^2 \tag{17}$$

subject to $\boldsymbol{B}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{I}_Q$, which corresponds to the Orthogonal Procrustes Problem (OPP) [37] with the solution

$$\boldsymbol{B}_k = \left(\boldsymbol{T}_k\boldsymbol{T}_k^{\mathrm{H}}\right)^{-\frac{1}{2}}\boldsymbol{T}_k, \tag{18}$$

where $\boldsymbol{T}_k = \boldsymbol{H}_k\widetilde{\boldsymbol{H}}_k$, and $\widetilde{\boldsymbol{H}}_k = \boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{B}_k = \boldsymbol{A}\boldsymbol{C}_k$. Next, we update the matrices $\boldsymbol{A}$ and $\boldsymbol{C}_k$ by solving (17) jointly for all ks. Let $\widetilde{\boldsymbol{\mathcal{H}}}$ be a tensor in which all $\widetilde{\boldsymbol{H}}_k$s are stacked as 3-mode slices. Then the constrained CP decomposition of $\widetilde{\boldsymbol{\mathcal{H}}}$ in terms of its frontal slices is written as

$$\widetilde{\boldsymbol{H}}_k = \boldsymbol{A}\boldsymbol{C}_k\boldsymbol{I}_Q. \tag{19}$$

Therefore, the 1-mode and the 3-mode unfoldings of $\widetilde{\boldsymbol{\mathcal{H}}}$ satisfy

$$[\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} = \boldsymbol{A}(\boldsymbol{I}_Q \diamond \boldsymbol{C})^{\mathrm{T}}, \tag{20}$$

$$[\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} = \boldsymbol{C}(\boldsymbol{A} \diamond \boldsymbol{I}_Q)^{\mathrm{T}}, \tag{21}$$

where the rows of $\boldsymbol{C}$ contain the diagonal elements of the matrices $\boldsymbol{C}_k$. Consequently, the least squares solution for the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ is calculated as follows

$$\boldsymbol{A} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)}(\boldsymbol{I}_Q \diamond \boldsymbol{C})^{\mathrm{T}+} \tag{22}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\mathrm{bdiag}\left\{ \frac{\boldsymbol{C}(:,1)^{\mathrm{H}}}{\|\boldsymbol{C}(:,1)\|^2}, \ldots, \frac{\boldsymbol{C}(:,\mathrm{Q})^{\mathrm{H}}}{\|\boldsymbol{C}(:,\mathrm{Q})\|^2} \right\} \right)^{\mathrm{T}}, \tag{23}$$

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)}(\boldsymbol{A} \diamond \boldsymbol{I}_Q)^{\mathrm{T}+} \tag{24}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[\operatorname{diag} \left\{ \frac{\boldsymbol{A}(1,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,1)\|^2} \right\}, \ldots, \operatorname{diag} \left\{ \frac{\boldsymbol{A}(Q,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,Q)\|^2} \right\} \right] \right)^{\mathrm{T}}. \tag{25}$$

Equations (23) and (25) allow us to avoid the explicit computation of the pseudo-inverse in (22) and (24), see Appendix for details.

In the original PARAFAC2 algorithm, the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ are computed by calculating the CP decomposition of $\tilde{\boldsymbol{\mathcal{H}}}$. In contrast to PARAFAC2, by taking into account the ML-GSVD constraints in (16), we can directly update the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ without computing the CP decomposition.

In the fourth step, a normalization of the columns of $\boldsymbol{C}$ is performed to ensure that they have unit norm. To ensure that the elements of $\boldsymbol{C}$ are real-valued, we multiply $\operatorname{diag}\{\boldsymbol{C}(k,:)\}$ by its complex conjugate, and compensate it in the columns of $\boldsymbol{A}$ in the fifth step. The algorithm stops if it exceeds the predefined maximum number of iterations or if the change of the residual given by $\Delta E_{\mathrm{R}} = \frac{(E_{\mathrm{R}}^{\mathrm{old}} - E_{\mathrm{R}})}{E_{\mathrm{R}}^{\mathrm{old}}}$ is smaller than a predefined threshold, implying the convergence of the algorithm, where $E_{\mathrm{R}}^{\mathrm{old}}$ represents the residual in the previous iteration. In the end of the algorithm, the elements of $\boldsymbol{C}$ are ordered as in (7) and in the description below this equation, while the columns of $\boldsymbol{A}$ and the rows of $\boldsymbol{B}_k$ are reordered accordingly.

A. Initializations and Computational Complexity

In the first case (described in the previous section), where $r_k = I$ for $\forall k$, the algorithm provides an essentially unique approximate solution up to a scaling of the columns in $\boldsymbol{A}$ and $\boldsymbol{B}_k$. It requires only one initialization of the algorithm. In the second case, we recommend to initialize the algorithm multiple times and choose the solution with the minimum reconstruction error E_R in order to avoid hitting a local minimum of the cost function (16). Given an arbitrary tensor, our simulations show that the proposed ML-GSVD is exact in Case 3. Taking into account the structure of the matrix $\boldsymbol{C}$ in (13), for Case 3, the algorithm can be directly initialized with the matrix of ones and zeros as in (13) ("closed form (CF)" initialization). Then, an update of $\boldsymbol{C}$ should be skipped in Step 3 of the algorithm. In Cases 1 and 2, a good approximation is obtained in the least squares sense. If $K = 2$, the ML-GSVD leads to an exact solution, and it is equal to the GSVD of two matrices.

The computational complexity per iteration of the main steps of the proposed Algorithm 1 is as follows. The SVD-based initialization would amount to $\mathcal{O}(I^3)$. The computational load to estimate the matrix $\boldsymbol{B}_k$ is $\mathcal{O}(KI^3)$. The direct LS solution (23) and (25) for the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ has complexity $\mathcal{O}(KI^2)$.

B. Alternative Ways to Update the Matrices $\boldsymbol{A}$ and $\boldsymbol{C}$

In this subsection, we consider three alternative ways to compute the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ in the third step of the Algorithm 1.

1) PARAFAC2 and Tensor Contractions: As previously mentioned, in general, any PARAFAC2 algorithm can be modified for the calculation of the ML-GSVD. Therefore, in the following we will show how the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ can be calculated via generalized tensor contractions [38]. In contrast to a slice-wise representation, the generalized tensor contractions allow to represent an N-way array explicitly in tensor form. Following the derivations in [38], we can express the tensor $\boldsymbol{\mathcal{H}} \in \mathbb{C}^{I \times K \times J}$ (the dimensions are permuted) in the constrained CP format

$$\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{I}}_{3,Q} \times_1 \bar{\boldsymbol{A}} \times_2 \bar{\boldsymbol{C}} \times_3 \bar{\boldsymbol{B}}, \tag{26}$$

which is equal to a constrained CP decomposition [39], where $\bar{\boldsymbol{A}} = \boldsymbol{A}(\boldsymbol{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_Q)$, $\bar{\boldsymbol{C}} = (\boldsymbol{I}_K \otimes \boldsymbol{1}_Q^{\mathrm{T}}) \diamond \operatorname{vec}\{\boldsymbol{C}^{\mathrm{T}}\}^{\mathrm{T}}$, and $\bar{\boldsymbol{B}} =$

Algorithm 1: ALS-Based Direct Fitting Algorithm for the Computation of the ML-GSVD of the Set of K Matrices $\boldsymbol{H}_k \in \mathbb{C}^{J_k \times I}$.

Require: Tensor $\boldsymbol{\mathcal{H}}$

1. Initialize $\boldsymbol{A}$ and $\boldsymbol{C}$

repeat

2. **for** $k = 1, 2 \ldots, K$ **do**

   Reconstruct $\widetilde{\boldsymbol{\mathcal{H}}}$ with $\boldsymbol{A}$ and $\boldsymbol{C}$:

   $$\widetilde{\boldsymbol{H}}_k = \boldsymbol{A} \cdot \operatorname{diag}\{\boldsymbol{C}(k,:)\}$$

   Update $\boldsymbol{B}_k$ $(k = 1, \ldots, K)$

   $$\boldsymbol{B}_k = (\boldsymbol{T}_k \boldsymbol{T}_k^{\mathrm{H}})^{-\frac{1}{2}} \boldsymbol{T}_k,\ \boldsymbol{T}_k = \boldsymbol{H}_k \widetilde{\boldsymbol{H}}_k$$

   Update $\widetilde{\boldsymbol{\mathcal{H}}}$: $\widetilde{\boldsymbol{H}}_k = \boldsymbol{H}_k^{\mathrm{H}} \cdot \boldsymbol{B}_k$

   end for

3. Update $\boldsymbol{A}$ and $\boldsymbol{C}$:

$$\boldsymbol{A} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\operatorname{bdiag} \left\{ \frac{\boldsymbol{C}(:,1)^{\mathrm{H}}}{\|\boldsymbol{C}(:,1)\|^2}, \ldots, \frac{\boldsymbol{C}(:,\mathrm{Q})^{\mathrm{H}}}{\|\boldsymbol{C}(:,\mathrm{Q})\|^2} \right\} \right)^{\mathrm{T}},$$

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[\operatorname{diag} \left\{ \frac{\boldsymbol{A}(1,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,1)\|^2} \right\}, \ldots, \operatorname{diag} \left\{ \frac{\boldsymbol{A}(Q,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,Q)\|^2} \right\} \right] \right)^{\mathrm{T}}$$

4. Normalize the columns of $\boldsymbol{C}$
5. To ensure the entries of $\boldsymbol{C}$ are real and non-negative, replace the columns of $\boldsymbol{C}$ by their absolute values, and compensate it in the columns of $\boldsymbol{A}$:

   for $i = 1, 2 \ldots, I$ **do**

   solve $|\boldsymbol{C}(:,i)| = \alpha \boldsymbol{C}(:,i)$, $\boldsymbol{A}(:,i) = \frac{1}{\alpha} \boldsymbol{A}(:,i)$

   end for

6. **for** $k = 1, 2 \ldots, K$ **do**

   $\hat{\boldsymbol{H}}_k = \boldsymbol{A} \cdot \operatorname{diag}\{\boldsymbol{C}(k,:)\} \cdot \boldsymbol{B}_k^{\mathrm{H}} \in \mathbb{C}^{I \times J_k}$

   end for

   $E_{\mathrm{R}} = (\|\hat{\boldsymbol{\mathcal{H}}} - \boldsymbol{\mathcal{H}}\|_{\mathrm{H}}^2) / \|\boldsymbol{\mathcal{H}}\|_{\mathrm{H}}^2$

until $\Delta E_{\mathrm{R}} = (E_{\mathrm{R}}^{\mathrm{old}} - E_{\mathrm{R}}) / E_{\mathrm{R}}^{\mathrm{old}}$ is smaller than a predefined threshold or the max number of iterations is reached. $E_{\mathrm{R}}^{\mathrm{old}}$ is the residual in the previous iteration

7. Order the columns of $\boldsymbol{C}$ as in (7), and reorder the columns of $\boldsymbol{A}$ and $\boldsymbol{B}_k$ accordingly.

$[\boldsymbol{\mathcal{B}}]_{(2)}$. Consequently, the matrix $\boldsymbol{A}$ can be estimated as follows

$$\boldsymbol{A} = [\boldsymbol{\mathcal{H}}]_{(1)} \cdot \left((\mathbf{1}_K^{\mathrm{T}} \otimes \boldsymbol{I}_Q) \cdot (\bar{\boldsymbol{B}} \diamond \bar{\boldsymbol{C}})^{\mathrm{T}} \right)^{+}. \tag{27}$$

Then the matrix $\boldsymbol{C}$ is computed by solving the following non-negative least squares problem

$$\mathrm{vec}\left\{\boldsymbol{\mathcal{H}}\right\} \approx \left([\widetilde{\boldsymbol{\mathcal{B}}}]_{(2)} \diamond (\boldsymbol{I}_K \otimes \mathbf{1}_R^{\mathrm{T}}) \diamond \bar{\boldsymbol{A}} \right) \cdot \mathrm{vec}\left\{\boldsymbol{C}^{\mathrm{T}}\right\}. \tag{28}$$

2) Least-Squares Khatri-Rao Factorization: The 2-mode unfolding of the tensor $\widetilde{\boldsymbol{\mathcal{H}}}$ in (19) can be written as follows

$$[\widetilde{\boldsymbol{\mathcal{H}}}]_{(2)} = (\boldsymbol{C} \diamond \boldsymbol{A})^{\mathrm{T}}, \tag{29}$$

which means that we can employ the least-squares Khatri-Rao factorization (LSKRF) [40] to estimate the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$.

3) Joint Diagonalization (JD): Let us consider the Gramian matrix $\boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{H}_k$

$$\boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{H}_k = \left(\boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} \right)^{\mathrm{H}} \left(\boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} \right) \tag{30}$$

$$= \boldsymbol{A} \boldsymbol{C}_k \boldsymbol{B}_k^{\mathrm{H}} \boldsymbol{B}_k \boldsymbol{C}_k \boldsymbol{A}^{\mathrm{H}} \tag{31}$$

$$= \boldsymbol{A} \boldsymbol{C}_k^2 \boldsymbol{A}^{\mathrm{H}} = \widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{H}}_k^{\mathrm{H}}. \tag{32}$$

The matrix $\boldsymbol{A}$ jointly diagonalizes the K Gramians $\boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{H}_k$ and therefore can be found from the following approximation problem [41]

$$\min \quad \sum_{k=1}^{K} \| \widetilde{\boldsymbol{H}}_k \widetilde{\boldsymbol{H}}_k^{\mathrm{H}} - \boldsymbol{A} \boldsymbol{\Lambda}_k \boldsymbol{A}^{\mathrm{H}} \|_{\mathrm{F}}^2, \tag{33}$$

where $\boldsymbol{\Lambda}_k = \boldsymbol{C}_k^2$ is a diagonal matrix. The author in [41] proposes the "AC-DC" algorithm that iteratively minimizes (33) alternating between the LS solution for the diagonalizing matrix $\boldsymbol{A}$ and the diagonal matrices $\boldsymbol{\Lambda}_k$. It should be noted that the solution of (33) is only essentially unique (permutation and scaling ambiguities are present), and the convergence of (33) does not guarantee the convergence of the parameters $\boldsymbol{A}$ and $\boldsymbol{\Lambda}_k$. However, we have observed in practice that this joint diagonalization provides satisfactory estimates for the matrices $\boldsymbol{A}$ and $\boldsymbol{C}_k$ and therefore can be used as a reliable starting point for the ML-GSVD algorithm.

V. Numerical Results

In this section, we carry out a simulation study of the ML-GSVD. In order to assess the performance of the proposed algorithm, we apply it to synthetic data. As an accuracy measure, we use the Squared Reconstruction Error (SRE)

$$\mathrm{SRE} = \frac{\left\| \hat{\boldsymbol{\mathcal{H}}} - \boldsymbol{\mathcal{H}} \right\|_{\mathrm{H}}^2}{\| \boldsymbol{\mathcal{H}} \|_{\mathrm{H}}^2}, \tag{34}$$

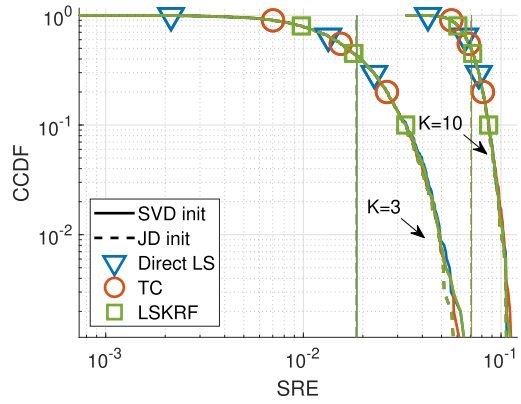
where $\boldsymbol{\mathcal{H}}$ is the original tensor, and $\hat{\boldsymbol{\mathcal{H}}}$ is the reconstructed tensor based on the estimated factor matrices $\boldsymbol{A}$, $\boldsymbol{B}_k$, and $\boldsymbol{C}_k$. In addition to the SRE, for each case of the ML-GSVD presented in Section III, the performance is also evaluated in terms of the average run time. In our simulations, a complex-valued tensor $\boldsymbol{\mathcal{H}}$ is generated randomly from a zero mean unit variance complex Gaussian distribution. The maximum number of iterations of the algorithm is set to 500. The algorithm stops when the change in the error is smaller than a predefined threshold (set to 10^{-5} in our simulations), or it reaches the maximum number of iterations. We compare the SVD- and JD-based initializations of the algorithm and different estimation schemes for the matrices $\boldsymbol{A}$ and $\boldsymbol{C}_k$: Direct LS solution ((23) and (25)), tensor contractions-based (TC), and the LSKRF-based solutions. For our simulations, we assume that the matrices in the decomposition have a full rank. The performances of the algorithms are represented by Complementary Cumulative Distribution Functions (CCDFs). In our simulations, the CCDF curves describe the probability that the SRE or the run time will be greater than a certain value on the x-axes. We present the CCDF of the error (time) since it allows an illustration of both average error (time) and its distribution. In all plots presented below the results were averaged over 1000 Monte Carlo runs. The vertical lines in the CCDF plots represent the mean values for each curve. Solid, dashed, and dotted lines represent SVD, JD, and CF initializations. Markers denote algorithms: blue triangle, red circle, and green square denote the direct least squares (Direct LS), tensor contractions (TC), and least squares Khatri-Rao factorization (LSKRF), respectively.

In Figs. 2(a) and 2(b) we depict the CCDF of the SRE and the CCDF of the execution time for Case 1 of the ML-GSVD, where the common column dimension is $I = 3$ and the row dimension J_k is equal to 4 for all 3-mode slices of the tensor. Fig. 2(a) depicts the SRE for tensors with $K = 3$ and $K = 10$ slices. As it can be observed, the reconstruction error increases with increasing K. This is also true for other cases of the ML-GSVD. Therefore, for the remaining simulations, we only display the results for $K = 3$ in order to avoid an overload of the plots. Although all initialization and factor estimation schemes display a similar SRE performance, the computational complexity varies. Fig. 2(b) shows that the Direct LS and LSKRF solutions have the smallest time complexity, and the solution using tensor contractions is the most complex from the computational point of view. This is explained by the fact that the first method does not require the computation of a matrix inverse, and the second solution involves the tensor unfolding and hence the multiplication of matrices of higher dimensions. If we configure the tensor as in Case 1, the ML-GSVD provides only a common subspace for all K matrices in the decomposition. Below is an example of the matrix $\boldsymbol{C}$ for Case 1:

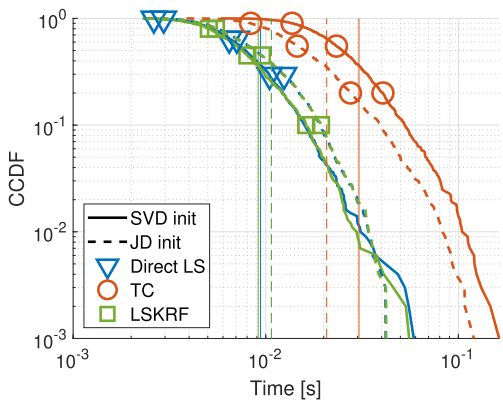
$$\boldsymbol{C} = \begin{bmatrix} \sigma_{1,1} & \cdots & \sigma_{1,I} \\ \vdots & \ddots & \vdots \\ \sigma_{K,1} & \cdots & \sigma_{K,I} \end{bmatrix} \in \mathbb{R}^{K \times I}, \tag{35}$$

where $\sigma_{k,i} \in (0, 1)$. We have observed in practice that this case requires only one initialization of the algorithm.

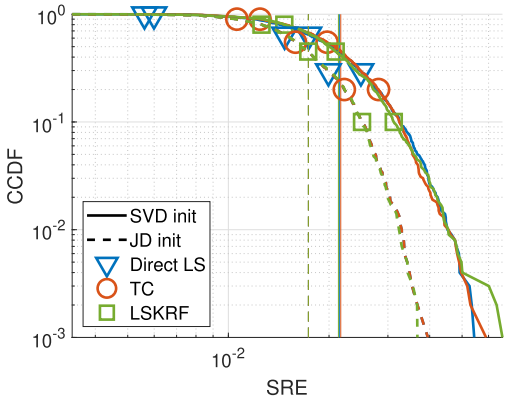
For the second case, the decomposition provides both common and private subspaces. Depending on the dimensions and ranks of the slices of the tensor, the private subspace can be empty for some ks. The common subspace can be shared by all the matrices in the decomposition or by some groups of matrices. Therefore, the second case is of great interest in terms of the applications. Below is an example of the matrix $\boldsymbol{C}$ for the Case



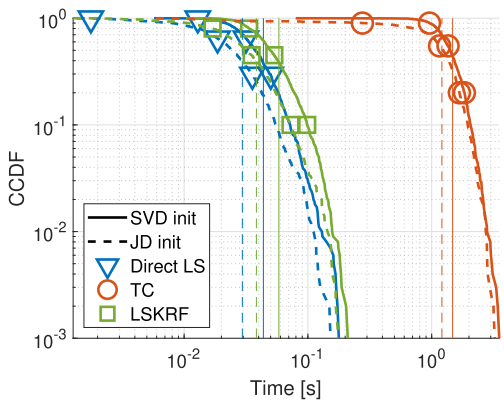
(a) CCDF vs. SRE. Case 1: $I = 3$, all $J_k = 4$.



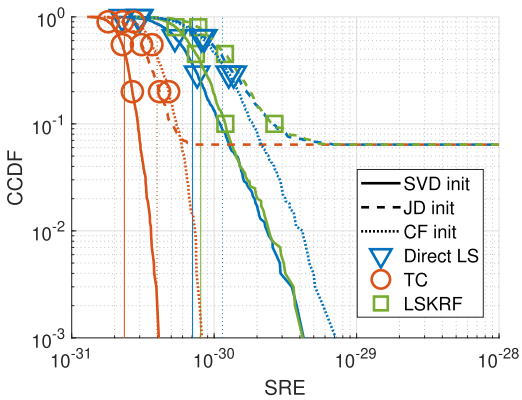
(b) CCDF vs. Time. Case 1: $I = 3, K = 3, J_k = 4$.



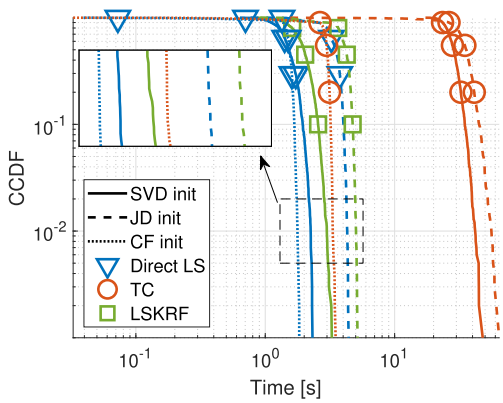
(c) CCDF vs. SRE. Case 2: $I = 8, K = 3, J_k = [3, 4, 4]$.



(d) CCDF vs. Time. Case 2: $I = 8, K = 3, J_k = [3, 4, 4]$.



(e) CCDF vs. SRE. Case 3: $I = 9, K = 3, J_k = [3, 3, 3]$.



(f) CCDF vs. Time. Case 3: $I = 9, K = 3, J_k = [3, 3, 3]$.

Fig. 2. CCDF vs. SRE and time for 3 cases of the ML-GSVD. Averaged over 1000 realizations of $\boldsymbol{\mathcal{H}}$. Solid, dashed, and dotted lines represent SVD, JD, and CF initializations. Markers denote algorithms: blue triangle, red circle, and green square denote the direct least squares (LS), tensor contractions (TC), and least squares Khatri-Rao factorization (LSKRF), respectively.

2 with $I = 8$ and the J_ks are equal to 3, 4, and 4, respectively

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sigma_{1,6} & \sigma_{1,7} & 1 \\ 0 & 0 & \sigma_{2,3} & 1 & 1 & \sigma_{2,6} & 0 & 0 \\ 1 & 1 & \sigma_{3,3} & 0 & 0 & 0 & \sigma_{3,7} & 0 \end{bmatrix} \in \mathbb{R}^{K \times I}. \quad (36)$$

As it can be observed from the third, sixth, and the seventh columns, the common subspace is shared by the groups of two matrices ($\{\sigma_{2,3}, \sigma_{3,3}\}$, $\{\sigma_{1,6}, \sigma_{2,6}\}$, and $\{\sigma_{1,7}, \sigma_{3,7}\}$). Moreover, all the matrices have a private subspace of dimension 1, 2, and 2, respectively. The CCDFs of the SRE and the time are shown in Figs. 2(c) and 2(d). In contrast to the previous results, for Case 2, the JD initialization leads to a smaller reconstruction error and converges faster.

Figs. 2(e) and 2(f) show the CCDFs of the SRE and the time for Case 3, where $I = 9$, and all $J_k = 3$, $K = 3$. The resulting

TABLE I
ALGORITHMS WITH THE SMALLEST RECONSTRUCTION ERROR AND THE SMALLEST TIME COMPLEXITY

	Case 1	Case 2	Case 3
Reconstruction error	all alg-s, any init.	all alg-s, JD init.	TC, SVD, CF init.
Time complexity	LSKRF, LS, SVD init.	LS, JD init.	LS, CF init.

matrix $\boldsymbol{C}$ has the following structure

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{K \times Q}. \quad (37)$$

Case 3 results in an exact decomposition with a private subspace for each slice of the tensor. We have observed in practice that in this case, the algorithm is prone to hit a local minimum. Therefore, for the SVD- and JD-based initializations, we recommend to use several initializations to ensure convergence. For the simulation results shown here, the algorithms were initialized 10 times. As it can be seen in Fig. 2(e), the closed form (CF) initialization converges in all runs, as well as the SVD-based initialization always converges to the exact solution because the matrix $\boldsymbol{C}$ is chosen randomly, and at least one initialization will lead to the global minimum. In Case 3 an initialization via JD is not recommended as it can be seen by the error floor in Fig. 2(e). The SVD initialization outperforms the JD in terms of the computational time, due to the iterative nature of the JD algorithm and the higher column dimension, compared to the row dimension. The probability of convergence also depends on the number of slices, and is higher for the smaller Ks.

Recommended implementations for the three cases: As it can be seen from Fig. 2(e), the CF initialization ensures the 100% convergence of the algorithm as compared to the JD-based initialization. Additionally, it reduces the time-complexity of the algorithm in comparison to SVD or JD-based initializations. Table I shows the algorithms and initializations that have the best reconstruction error and the smallest time complexity performances. Considering the simulation results and the accuracy-complexity trade-off, we recommend to use the Direct LS solution with one SVD-based initialization for Case 1, multiple JD-initializations for Case 2, and Direct LS with closed form initialization for Case 3.

In the next simulation, we assess how the simulation time changes with increasing K (number of tensor slices). We use the SVD-based initialization and different ways to estimate the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$. As it can be observed in Fig. 3, the computational time increases with the increasing K for the tensor contractions based estimation, but does not change greatly in case of the direct LS or LSKRF-based solutions.

VI. APPLICATION OF THE ML-GSVD IN MULTICAST BEAMFORMING

As a promising application of the proposed ML-GSVD, we consider coordinated beamforming using joint broadcast and

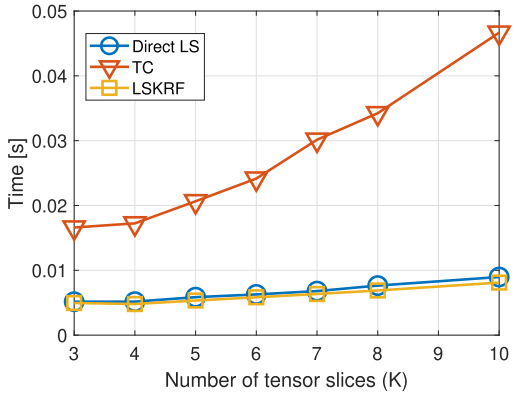


Fig. 3. Simulation time. The third dimension (K) is changing, $I = 3$, $J_k = 3$.

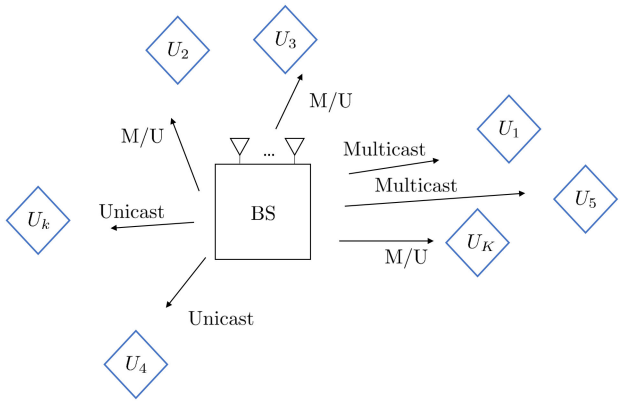


Fig. 4. Joint multicast and unicast transmission.

unicast transmissions. We view the 3-mode slices of the tensor $\boldsymbol{\mathcal{H}}$ as the channels of K users and construct the precoding and decoding matrices based on the ML-GSVD factor matrices. In the following, we will describe a simple "toy" model of the ML-GSVD-based communication system and show how the subspace structure of the ML-GSVD, presented in Section III, can be utilized to simultaneously send the common messages to a group of users and private messages to individual users.

A. System Model and ML-GSVD-Based Beamforming

We consider a downlink MIMO-OFDM system with multicast transmission as depicted in Fig. 4, where one base station (BS) equipped with M_T antennas transmits *common* and *private* messages to K users with M_{R_k} receive antennas each using N subcarriers. We represent the MIMO-OFDM channel between the BS and the kth user on the nth subcarrier by $\boldsymbol{H}_{k,n} \in \mathbb{C}^{M_{R_k} \times M_T}$. We focus on the case where $M_{R_k} < M_T$, and $\sum_{k=1}^{K} M_{R_k} > M_T$. This scenario corresponds to the case where the ML-GSVD provides both private and common subspaces (Case 2). Furthermore, each user observes zero mean circularly symmetric complex Gaussian white noise $\boldsymbol{n}_{k,n}$ with variance σ_n^2. Then, the received signal on subcarrier n of the kth user is given by

$$\boldsymbol{y}_{k,n} = \boldsymbol{H}_{k,n}\boldsymbol{F}_n\boldsymbol{x}_n + \boldsymbol{n}_{k,n}, \quad (38)$$

and at the detector we get

$$\hat{\boldsymbol{y}}_{k,n} = \boldsymbol{W}_{k,n}\left(\boldsymbol{H}_{k,n}\boldsymbol{F}_n\boldsymbol{x}_n + \boldsymbol{n}_{k,n}\right), \tag{39}$$

where $\boldsymbol{x}_n$ denotes the transmitted signal on the nth subcarrier and satisfies $\mathbb{E}\{\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{H}}\} = \boldsymbol{I}_{M_T}/M_T$. The matrices $\boldsymbol{F}_n$ and $\boldsymbol{W}_{k,n}$ are the common transmit beamforming and the receive beamforming matrices, respectively. As in (5), the ML-GSVD of the channel matrices $\boldsymbol{H}_{k,n}$ is given by $\boldsymbol{H}_{k,n} = \boldsymbol{B}_{k,n}\boldsymbol{C}_{k,n}\boldsymbol{A}_n^{\mathrm{H}}$. Therefore, the precoding matrix $\boldsymbol{F}_n$ can be determined as $\boldsymbol{F}_n = \alpha\{\boldsymbol{A}_n^{\mathrm{H}}\}_{\mathcal{C}(\mathcal{Q})}^{-1}$, and the receive decoding matrices as $\boldsymbol{W}_{k,n} = \{\boldsymbol{B}_{k,n}^{\mathrm{H}}\}_{\mathcal{R}(\mathcal{Q})}$, where α is a power normalization coefficient, and $\mathcal{C}(\mathcal{Q})$ and $\mathcal{R}(\mathcal{Q})$ denote columns and rows of the matrix with indices in the set $\mathcal{Q} \subseteq \{1, \ldots, M_T\}$. An appropriate selection scheme will be described at the end of this subsection. The elements of $\boldsymbol{C}_k$ contain the normalized gains of the corresponding virtual channels (VCs) (private or common). The condition $M_T > M_{R_k}$ is the requirement to have private subspaces, while if $M_T \leq M_{R_k}$ only broadcasting is possible. The private subspaces (ones and zeros) in $\boldsymbol{C}_k$ are used by the transmitter to send confidential messages to the user U_k, while the common subspace ($\boldsymbol{\sigma}_k$ in $\boldsymbol{C}_k$) is used for broadcasting common messages to all users. The private subspaces always have unit normalized gains, while the normalized gains of the common subspaces are less than one. Note that the resulting number of private and common subspaces depends both on the dimensionality and realization of the channel tensor $\boldsymbol{\mathcal{H}}_n \in \mathbb{C}^{M_R \times M_T \times K}$, $M_R = \max(M_{R_1}, \ldots, M_{R_K})$.

The matrices $\boldsymbol{A}_n$ and $\boldsymbol{B}_{k,n}$ jointly diagonalize the channel represented by $\boldsymbol{H}_{k,n}$ to get virtual channels that enable a simultaneous point-to-multi-point connection with private and common messages. The required subset of VCs (private or common) can be chosen by an appropriate selection of the columns of the transmit precoding matrix and the corresponding rows of the receive beamforming matrices. For instance, if the ith and $(i+1)$th columns of $\boldsymbol{C}$ lie in a common subspace, for broadcasting, we choose $\mathcal{Q} \in \{i, i+1\}$, and select the ith and $(i+1)$th columns and rows of the transmit and receive beamforming matrices, respectively. For the transmission of the private messages, we choose the vectors that correspond to a private subspace (values equal to one in $\boldsymbol{C}$), and thus, enabling a simultaneous transmission of private and common messages. If multiple private subspaces are available, we transmit multiple streams to that particular user. We construct the channel tensor $\boldsymbol{\mathcal{H}}_n$ for each subcarrier and calculate the ML-GSVD of the set of K matrices to obtain the precoding and decoding matrices on each subcarrier.

B. Simulation Results

In this subsection, we evaluate the performance of the proposed ML-GSVD-based beamforming in terms of the achievable sum rates and compare it with the state-of-the-art techniques. To the best of our knowledge, there are no schemes in the literature that combine MIMO-OFDM and multicast/unicast transmission. Therefore, when assessing the performance of the multicast transmission, we compare it to the algorithm in [20], and in case of the unicast transmission we use the FlexCoBF [42], [43] scheme as a reference.

The authors in [20] propose a multicast precoding scheme for MIMO-OFDM system to transmit only common messages to a selected group of users. They employ the linear sum (LS) based precoding matrix, which is given as follows

$$\boldsymbol{F}_n^{\mathrm{LS}} = \alpha \sum_{k \in \mathcal{S}} \beta_k \boldsymbol{V}_{n,k}, \tag{40}$$

to maximize the individual rate of user k on the subcarrier n. The matrix $\boldsymbol{V}_{n,k}$ is the right singular matrix corresponding to the non-zero singular values of the channel matrix $\boldsymbol{H}_{n,k}$, $\mathcal{S}$ is a subset of users that receive the common message, β_k is a power normalization according to the effective channel gain, and α is a normalization constant to fulfill the power constraint

$$\left\|\boldsymbol{F}_n^{\mathrm{LS}}\right\|_F^2 = P_n. \tag{41}$$

We compare this linear sum based precoding in (40) with the ML-GSVD-based precoding

$$\boldsymbol{F}_n = \alpha\{\boldsymbol{A}_n^{\mathrm{H}}\}_{\mathcal{C}(\mathcal{Q})}^{-1}, \tag{42}$$

where $\boldsymbol{A}_n$ is obtained from the ML-GSVD of the channel tensor $\boldsymbol{\mathcal{H}}_n$, and $\mathcal{C}(\mathcal{Q}_{\mathcal{S}})$ is the set of columns that correspond to a common subspace of the group of users $\mathcal{S}$.

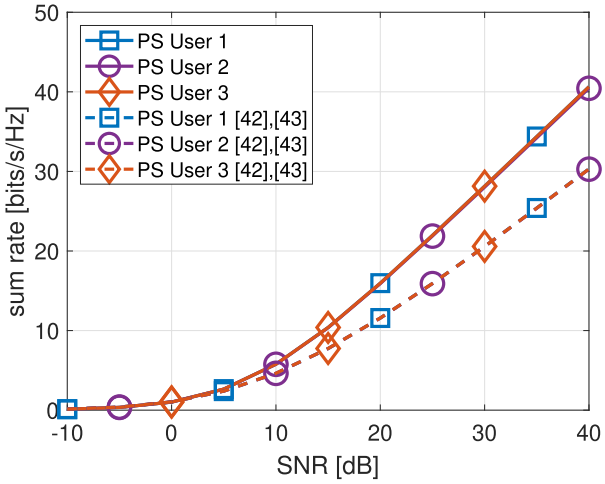
For the private messages transmission, we compare the ML-GSVD-based scheme with FLexCoBF proposed in [42], [43]. The FlexCoBF technique is applicable in the case where the total number of receive antennas of the served users is larger than the number of transmit antennas of the serving base station (BS). For more details, we refer the reader to [42], [43].

We compare the performance of the different algorithms in terms of the sum rate, and calculate the achievable rate of the user k on subcarrier n as follows

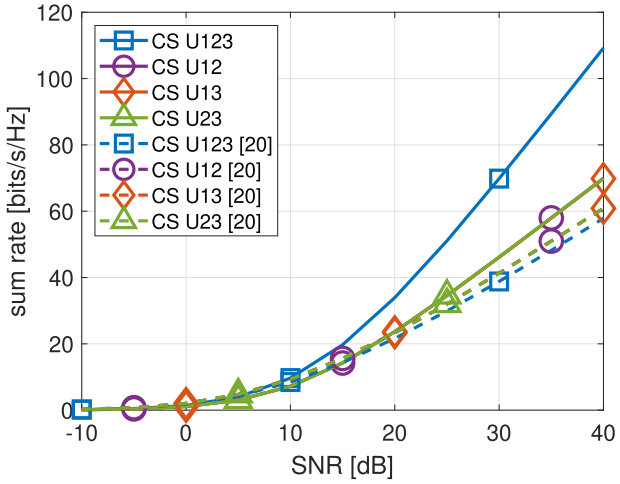
$$\mathrm{R}_{k,n} = \log_2 \det\left(\boldsymbol{I}_{N_r} + \frac{P_n}{M_T N_0}\boldsymbol{H}_{n,k}\boldsymbol{F}_n\boldsymbol{F}_n^{\mathrm{H}}\boldsymbol{H}_{n,k}^{\mathrm{H}}\right). \tag{43}$$

In contrast to the aforementioned schemes, the ML-GSVD enables the simultaneous transmission of common and private messages. In the following, we will consider two simulation scenarios to assess the performance of the ML-GSVD scheme in terms of the achievable sum rate. Furthermore, we will compare it with the aforementioned schemes.

For the first simulation, we consider a MIMO-OFDM system with 5 uncorrelated subcarriers and $K = 3$ users with 6 antennas each. The number of transmit antennas at the base station is equal to $M_T = 9$. Such a configuration of the system enables both unicast and multicast transmission (see Section III). To illustrate the effect of simultaneously serving common and private channels via the ML-GSVD, we consider a simple uncorrelated Rayleigh fading MIMO channel model. With the ML-GSVD-based scheme we can transmit both common and private messages simultaneously on the same subcarrier as it is shown in Table II, while for the reference algorithms we use some subcarriers for the unicast and some for the multicast transmission as it is shown in Table II, where p_1, p_2, and p_3 denote the private channels to the corresponding users, and c_{123}, c_{12}, c_{13}, and c_{23} are the common channels to all three



(a) Sum rate vs. SNR. Private messages to three users.



(b) Sum rate vs. SNR. Common messages to groups of 2 and 3 users.

Fig. 5. Multicast transmission: Private (PS) and common (CS) subspaces. 3 users, $M_T = 9$, $M_R = [6, 6, 6]$. Averaged over 5 subcarriers and 10000 trials. The solid lines represent the proposed ML-GSVD precoding, and the dashed lines correspond to the reference algorithms.

TABLE II
JOINT MULTICAST AND UNICAST TRANSMISSION: 5 SUBCARRIERS AND 3 USERS (ML-GSVD & REFERENCE ALGORITHMS)

Proposed scheme			
N	Channels	# of streams r_k	Algorithm
1	p_k & $c_{\{l\}}$	p_k: $\{0 \dots M_{R_k}\}$	ML-GSVD
2	p_k & $c_{\{l\}}$	$c_{\{l\}}$: $\{0 \dots \min\{M_{R_{\{l\}}}\}\}$	ML-GSVD
3	p_k & $c_{\{l\}}$	for each subcarrier,	ML-GSVD
4	p_k & $c_{\{l\}}$	$\sum_{n=1}^{N} r_{k,n}$ in total	ML-GSVD
5	p_k & $c_{\{l\}}$	$\{l\} \subseteq \{1, \dots, K\}$	ML-GSVD
State-of-the-art			
1	p_1, p_2, p_3	r_1, r_2, r_3	FlexCoBF [42], [43]
2	c_{123}	$\min\{M_{R_k}\}$	Linear Sum [20]
3	c_{12}	$\min\{M_{R_1}, M_{R_2}\}$	Linear Sum [20]
4	c_{13}	$\min\{M_{R_1}, M_{R_3}\}$	Linear Sum [20]
5	c_{23}	$\min\{M_{R_2}, M_{R_3}\}$	Linear Sum [20]

users, and to two selected users, respectively. Note that the ML-GSVD enables the automatic user grouping and scheduling based on the ML-GSVD singular values (matrix $\boldsymbol{C}$ in (5)). On the other hand, for the reference algorithms, we have to fix the unicast and multicast groups and assign them to different subcarriers. Furthermore, we also fix the number of streams r_k for the different users and channels as it is shown in Table II. For the reference algorithms, we set the number of private streams to each user to $r_k = 3$, where $\sum_{k=1}^{K} r_k = M_T$, and the number of common streams is shown in Table II. In case of the ML-GSVD-based beamforming, the number of virtual channels (VCs) depends on the realization of the channel. The number of private and common streams varies for each subcarrier: $c_{\{l\}} = \{0 \dots \min\{M_{R_{\{l\}}}\}\}$, where $\{l\} \subseteq \{1, \dots, K\}$. We calculate the minimum achievable sum rate for each type of channel (common or private) as $\min_k\{\sum_{n=1}^{N} R_{k,n}\}$ and average the results over $T = 10000$ Monte-Carlo trials.

Fig. 5(a) depicts the sum rate of the private channels to all three users and Fig. 5(b) shows the average sum rate for common channels to a group of two users or to all three users. The solid and dashed lines correspond to the ML-GSVD-based

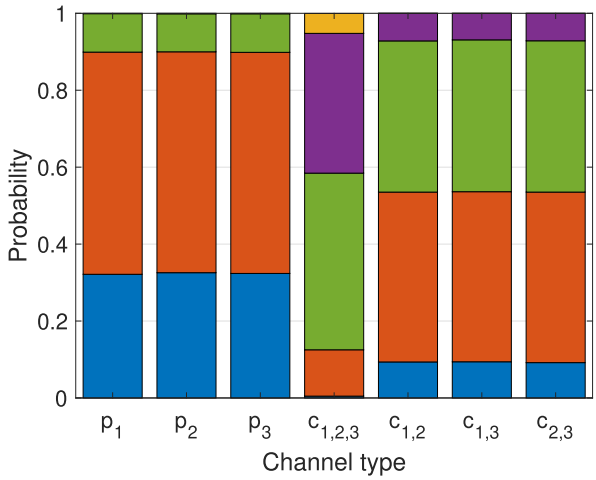
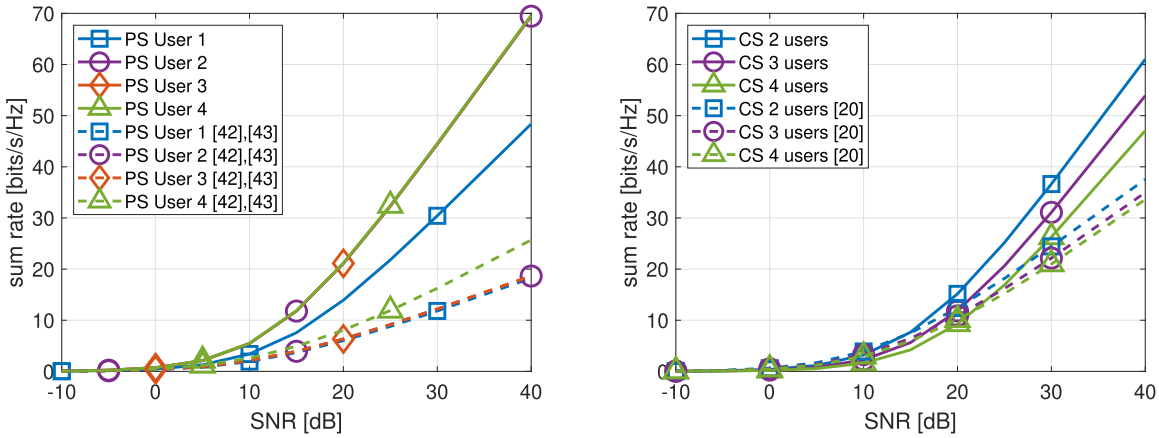


Fig. 6. Histogram - Private and common channels (ML-GSVD). Colors indicate the average number of VCs (streams) at each subcarrier: -0, -1, -2, -3, and -4 VCs, respectively. Labels on the x-axis of the histogram represent the users or the groups of users (p_1 is a private channel to user 1, $c_{1,2}$ is a common channel to users 1 and 2, etc.).

and reference schemes-based beamforming, respectively. As it can be observed, the ML-GSVD-based scheme outperforms the reference schemes for both private and common messages. This is explained by the fact that the ML-GSVD precoding allows transmitting both private and common messages on one subcarrier, which increases the average total number of streams to each user, which can be calculated as follows

$$\hat{r}_k = \frac{1}{T} \sum_{n=1}^{N} s_{k,n}, \tag{44}$$

where $s_{k,n}$ is the number of streams for the given channel type, N is the number of subcarriers, and T is the number of simulation trials. Additionally, the proposed ML-GSVD-based scheme does not fix the number of private and common streams,



(a) Sum rate vs. SNR. Private messages.

(b) Sum rate vs. SNR. Common messages to sets of 2, 3, and 4 users.

Fig. 7. Multicast transmission: Private (PS) and common (CS) subspaces. 4 users, $M_T = 9$, $M_R = [4, 5, 5, 5]$, $r_k = [2, 2, 2, 3]$. Averaged over 12 subcarriers and 10000 trials. The solid lines represent the proposed ML-GSVD precoding, and the dashed lines correspond to the reference algorithms.

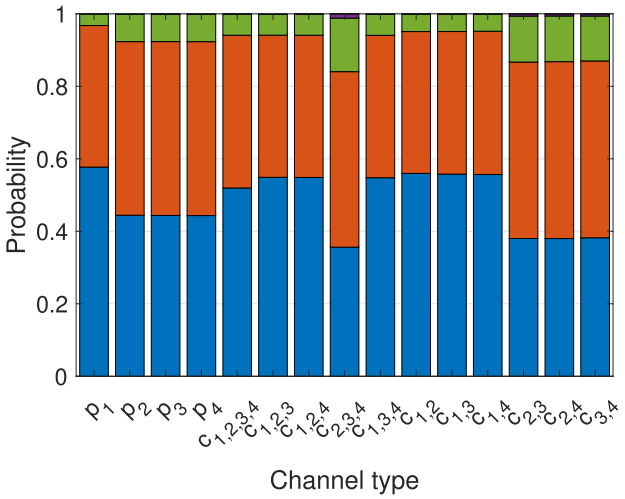


Fig. 8. Histogram - Private and common channels (ML-GSVD). Colors indicate the average number of VCs (streams) at each subcarrier: -0, -1, -2, and -3 VCs, respectively. Labels on the x-axis of the histogram represent users or groups of users (p_1 is a private channel to user 1, $c_{1,2}$ is a common channel to users 1 and 2, etc.).

and their number depends on the realization of the channel and, therefore, can vary depending on the channel conditions. As a result, we observe a higher sum rate for most of the channel types. The rates for the common channels are higher than the rates of the private channels due to the fact that, as can be seen on the histogram, there are on the average more common channels than private channels.

Fig. 6 illustrates the histogram of the probability of having private subspaces (PSs) and common subspaces (CSs) for each user or user group. Labels on the x-axis represent the users or the groups of users (p_1 is a private channel to user 1, $c_{1,2}$ is a common channel to users 1 and 2, etc.). The results are averaged over all subcarriers. As it can be seen from the histogram, the proposed ML-GSVD-based beamforming provides CSs between all three users and CSs between two of three users. As described in Section III, the number of these subspaces (virtual channels) depends on the realization of the channel, the number of receive and transmit antennas (tensor dimensions), and the channel rank. In case of two users, the number of common virtual channels is equal to $M_{R_1} + M_{R_2} - M_T$ and number of private channels is $M_T - M_{R_2}$ and $M_T - M_{R_1}$, respectively (assuming full-rank channels). While for two users (where the ML-GSVD simplifies to the GSVD), the dimensions of PSs and CSs can be described explicitly, with $K > 2$ users, additional subspaces between the subgroups of users can appear, which highly depends on the realization of the channel. With an increasing number of users and antennas, more combinations of private and common subspaces are possible. The histograms demonstrate the probabilities for the expected dimensions of those particular channels. As it can be observed from Fig. 6, in this simulation scenario users can have 0, 1, or 2 private channels (1 with the highest probability). There is also a high probability to obtain 1 or 2 common channels to a group of two users, and 2 or 3 common channels to all users. The total number of virtual channels (streams) for one realization of the channel tensor is equal to M_T. Therefore, for some of the Monte-Carlo runs, there are zero channels of a certain type. This can be associated, for instance, with bad channel conditions and has a potential to be used for the selection of the optimal set of users to be served.

In contrast to the first simulation, in the next simulation we consider an asymmetric scenario with the same number of transmit antennas $M_T = 9$ but an increased number of $K = 4$ users having 4, 5, 5, and 5 antennas, respectively. This configuration also leads to both private and common virtual channels. Due to the increased number of users, the number of possible channel types also increases. Therefore, in this scenario, we assume 12 subcarriers. As in the previous simulation, we transmit private and common messages simultaneously on all subcarriers based on the ML-GSVD. For the reference algorithms, one subcarrier is used for the private messages, and the other subcarriers are employed for the transmission of common messages, similarly as in Table II. There are 15 types of virtual channels in total, which includes private channels p_1, p_2, p_3, and p_4, common

channels to all four users c_{1234}, common channels to three of four users c_{123}, c_{124}, c_{234}, and c_{134} (combinations of 3 users out of a set of 4 users which is described by the binomial coefficient $\binom{4}{3}$), and common channels to two of four users c_{12}, c_{13}, c_{14}, c_{23}, c_{24}, and c_{23} (combinations of 2 users out of a set of 4 users, described by the binomial coefficient $\binom{4}{2}$). Fig. 7(a) depicts the achievable sum rates for the private channels of four users. The solid lines represent the proposed scheme, and the dashed lines denote the FlexCoBF precoding scheme from [42] and [43]. Fig. 7(b) depicts the average sum rates for the common channels to a group of two, three, and four users. As in the previous simulation scenario, the ML-GSVD scheme (solid lines) is compared to the multicast beamforming technique in [20] (dashed lines). As it can be seen, the proposed scheme outperforms the reference techniques for all channel types. Fig. 8 shows the possible channel types associated with this simulation scenario and their corresponding probabilities in case of the ML-GSVD-based transmission.

We have also observed in practice that with an increasing number of users K, the probability to obtain a common channel to all users decreases, and there is a higher chance to have private channels or common channels to a group of two or three users. This also opens up an opportunity to use the ML-GSVD for the user scheduling, since it can be performed automatically based on the generalized singular values of the ML-GSVD.

VII. Conclusion

We have presented a new Multilinear Generalized Singular Value Decomposition (ML-GSVD) as an extension of the matrix-based GSVD to jointly factorize a set of an arbitrary number ($K \geq 2$) of matrices with a common number of rows or columns. To this end, we have proposed algorithms to compute the ML-GSVD as a tensor factorization with constraints. In comparison with existing GSVD generalizations, our ML-GSVD preserves the properties of the original GSVD, such as orthogonality of the 2-mode factor matrices. We have considered three different cases of the ML-GSVD depending on the dimensions and the rank-structure of the decomposed matrices. Moreover, we have shown that every case provides a certain decomposition structure with private and/or common subspaces. An ALS-based algorithm to compute the ML-GSVD has been developed as an extension of an algorithm to compute the PARAFAC2 decomposition. Furthermore, we have specified an appropriate initialization scheme for each case of the ML-GSVD to guarantee convergence.

As an application of the proposed ML-GSVD, we have considered multiuser MIMO-OFDM systems with joint unicast and multicast transmissions. For such a scenario, we have shown that the factor matrices of the ML-GSVD can be used as precoding and decoding matrices, respectively. Moreover, we demonstrate that the aforementioned three cases of the ML-GSVD correspond to the transmission of private or common messages (or both) to different sets of users. According to our numerical results, the ML-GSVD outperforms the reference multicast and unicast precoding schemes in terms of the average sum rate.

Since the ML-GSVD extends the matrix GSVD, it can be employed further in a number of different communication and biomedical applications, such as coordinated beamforming, MIMO relaying, physical layer security, and genomic signal processing. The ML-GSVD allows to increase the number of jointly factorized matrices and, therefore, can be used in more complex systems.

Appendix

Exploiting the Khatri-Rao product with an identity matrix, the least squares solution for the matrix $\boldsymbol{A}$ in (22), can be expressed as

$$\boldsymbol{A} = \left[\widetilde{\boldsymbol{\mathcal{H}}}\right]_{(1)} \left(\mathrm{bdiag}\left\{\boldsymbol{C}(:,1),\ldots,\boldsymbol{C}(:,Q)\right\}\right)^{\mathrm{T}+} \tag{45}$$

Then, using the Moore–Penrose pseudo-inverse of the block-diagonal matrix, we can rewrite (45) as follows

$$\boldsymbol{A} = \left[\widetilde{\boldsymbol{\mathcal{H}}}\right]_{(1)} \Bigg(\left[1 \oslash \mathrm{diag}\left\{\|\boldsymbol{C}(:,1)\|^2,\ldots,\|\boldsymbol{C}(:,Q)\|^2\right\}\right] \tag{46}$$

$$\cdot\, \mathrm{bdiag}\left\{\boldsymbol{C}(:,1),\ldots,\boldsymbol{C}(:,Q)\right\}^{\mathrm{H}} \Bigg)^{\mathrm{T}} \tag{47}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(1)} \left(\mathrm{bdiag}\left\{\frac{\boldsymbol{C}(:,1)^{\mathrm{H}}}{\|\boldsymbol{C}(:,1)\|^2},\ldots,\frac{\boldsymbol{C}(:,\mathrm{Q})^{\mathrm{H}}}{\|\boldsymbol{C}(:,\mathrm{Q})\|^2}\right\}\right)^{\mathrm{T}}, \tag{48}$$

where $\oslash$ denotes the element-wise division. With the identity on the right hand side of the Khatri-Rao product, equation (24) can be rewritten as follows

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\begin{bmatrix}\mathrm{diag}\left\{\boldsymbol{A}(1,:)\right\}\\ \vdots \\ \mathrm{diag}\left\{\boldsymbol{A}(Q,:)\right\}\end{bmatrix}\right)^{\mathrm{T}+}. \tag{49}$$

Again, using the Moore–Penrose pseudo-inverse, the least squares solution for $\boldsymbol{C}$ can be expressed as

$$\boldsymbol{C} = [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \Bigg(\left[1 \oslash \mathrm{diag}\left\{\|\boldsymbol{A}(:,1)\|^2,\ldots,\|\boldsymbol{A}(:,Q)\|^2\right\}\right] \tag{50}$$

$$\cdot \begin{bmatrix}\mathrm{diag}\left\{\boldsymbol{A}(1,:)\right\}\\ \vdots \\ \mathrm{diag}\left\{\boldsymbol{A}(Q,:)\right\}\end{bmatrix}^{\mathrm{H}} \Bigg)^{\mathrm{T}} \tag{51}$$

$$= [\widetilde{\boldsymbol{\mathcal{H}}}]_{(3)} \left(\left[\mathrm{diag}\left\{\frac{\boldsymbol{A}(1,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,1)\|^2}\right\},\ldots,\mathrm{diag}\left\{\frac{\boldsymbol{A}(Q,:)^{\mathrm{H}}}{\|\boldsymbol{A}(:,Q)\|^2}\right\}\right]\right)^{\mathrm{T}}. \tag{52}$$

REFERENCES

[1] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "Multilinear generalized singular value decomposition (ML-GSVD) with application to coordinated beamforming in multi-user MIMO systems," in *Proc. 45th IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 4587–4591.

[2] S. A. A. Fakoorian and A. L. Swindlehurst, "Optimal power allocation for GSVD-based beamforming in the MIMO Gaussian wiretap channel," in *Proc. IEEE Int Symp. Inf. Theory*, 2012, pp. 2321–2325.

[3] W. Mei, Z. Chen, and J. Fang, "GSVD-based precoding in MIMO systems with integrated services," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1528–1532, Nov. 2016.

[4] D. Hassan, S. Redif, J. G. McWhirter, and S. Lambotharan, "Polynomial GSVD beamforming for two-user frequency-selective MIMO channels," *IEEE Trans. Signal Process.*, vol. 69, pp. 948–959, 2021.

[5] D. Senaratne and C. Tellambura, "Generalized singular value decomposition for coordinated beamforming in MIMO systems," in *Proc. IEEE Glob. Telecommun. Conf.*, 2010, pp. 1–6.

[6] D. Senaratne and C. Tellambura, "GSVD beamforming for two-user MIMO downlink channel," *IEEE Trans. Veh. Technol.*, vol. 62, no. 6, pp. 2596–2606, Jul. 2013.

[7] A. Khisti, G. Wornell, A. Wiesel, and Y. Eldar, "On the Gaussian MIMO wiretap channel," in *Proc. IEEE Int. Symp. Inf. Theory*, 2007, pp. 2471–2475.

[8] Y. Wu, J. Wang, J. Wang, R. Schober, and C. Xiao, "Secure transmission with large numbers of antennas and finite alphabet inputs," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3614–3628, Aug. 2017.

[9] K. A. Aiello and O. Alter, "Platform-independent genome-wide pattern of DNA copy-number alterations predicting astrocytoma survival and response to treatment revealed by the GSVD formulated as a comparative spectral decomposition," *PLoS One*, vol. 11, no. 10, pp. 1–22, Oct. 2016. [Online]. Available: https://doi.org/10.1371/journal.pone.0164546

[10] O. Alter, P. O. Brown, and D. Botstein, "Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms," *Proc. Nat. Acad. Sci.*, vol. 100, no. 6, pp. 3351–3356, Mar. 2003. [Online]. Available: https://www.pnas.org/content/100/6/3351

[11] C. H. Lee, B. O. Alpert, P. Sankaranarayanan, and O. Alter, "GSVD comparison of patient-matched normal and tumor aCGH profiles reveals global copy-number alterations predicting glioblastoma multiforme survival," *PLoS One*, vol. 7, no. 1, pp. 1–11, Jan. 2012. [Online]. Available: https://doi.org/10.1371/journal.pone.0030098

[12] C. Paige and M. Saunders, "Towards a generalized singular value decomposition," *SIAM J. Numer. Anal.*, vol. 18, no. 3, pp. 398–405, Jun. 1981. [Online]. Available: https://doi.org/10.1137/0718026

[13] C. F. Van Loan, "Generalizing the singular value decomposition," *SIAM J. Numer. Anal.*, vol. 13, pp. 76–83, Mar. 1976.

[14] S. P. Ponnapalli, M. A. Saunders, C. F. Van Loan, and O. Alter, "A higher-order generalized singular value decomposition for comparison of global mRNA expression from multiple organisms," *PLoS One*, vol. 6, no. 12, pp. 1–11, Dec. 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0028072

[15] P. Sankaranarayanan, T. E. Schomay, K. A. Aiello, and O. Alter, "Tensor GSVD of patient- and platform-matched tumor and normal DNA copy-number profiles uncovers chromosome arm-wide patterns of tumor-exclusive platform-consistent alterations encoding for cell transformation and predicting ovarian cancer survival," *PLoS One*, vol. 10, no. 4, pp. 1–21, Apr. 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0121396

[16] I. Kempf, P. J. Goulart, and S. R. Duncan, "A higher-order generalized singular value decomposition for rank deficient matrices," Jul. 2021, *arXiv:2102.09822*.

[17] Z. Chen, Z. Ding, X. Dai, and R. Schober, "Asymptotic performance analysis of GSVD-NOMA systems with a large-scale antenna array," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 575–590, Jan. 2019.

[18] C. Rao, Z. Ding, and X. Dai, "The distribution characteristics of ordered GSVD singular values and its applications in MIMO-NOMA," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2719–2722, Dec. 2020.

[19] L. Khamidullina, A. L. F. de Almeida, and M. Haardt, "ML-GSVD-based MIMO-NOMA networks," in *Proc. 25th Int. ITG Workshop Smart Antennas*, 2021, pp. 1–6.

[20] J. Joung, H. D. Nguyen, P. H. Tan, and S. Sun, "Multicast linear precoding for MIMO-OFDM systems," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 993–996, Jun. 2015.

[21] N. Jindal and Z. Luo, "Capacity limits of multiple antenna multicast," in *Proc. IEEE Int. Symp. Inf. Theory*, 2006, pp. 1841–1845.

[22] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.

[23] I. H. Kim, D. J. Love, and S. Y. Park, "Optimal and successive approaches to signal design for multiple antenna physical layer multicasting," *IEEE Trans. Commun.*, vol. 59, no. 8, pp. 2316–2327, Aug. 2011.

[24] H. Zhu, N. Prasad, and S. Rangarajan, "Precoder design for physical layer multicasting," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5932–5947, Jun. 2012.

[25] Z. Xiang, M. Tao, and X. Wang, "Coordinated multicast beamforming in multicell networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 1, pp. 12–21, Jan. 2013.

[26] J. Yu and M. Dong, "Distributed low-complexity multi-cell coordinated multicast beamforming with large-scale antennas," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun.*, 2018, pp. 1–5.

[27] Z. Yang and M. Dong, "Low-complexity coordinated relay beamforming design for multi-cluster relay interference networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2215–2228, Apr. 2019.

[28] T. X. Doan, H. Q. Ngo, T. Q. Duong, and K. Tourki, "On the performance of multigroup multicast cell-free massive MIMO," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2642–2645, Dec. 2017.

[29] M. S. Ibrahim, A. Konar, and N. D. Sidiropoulos, "Fast algorithms for joint multicast beamforming and antenna selection in massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 1897–1909, Mar. 2020.

[30] M. Sadeghi, E. Björnson, E. G. Larsson, C. Yuen, and T. Marzetta, "Joint unicast and multi-group multicast transmission in massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6375–6388, Oct. 2018.

[31] Y. C. B. Silva and A. Klein, "Linear transmit beamforming techniques for the multigroup multicast scenario," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4353–4367, Oct. 2009.

[32] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, Mar. 2000. [Online]. Available: https://doi.org/10.1137/S0895479896305696

[33] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2013.

[34] Z. Bai, "The CSD, GSVD, Their Applications and Computations," . Univ. Minnesota, Minneapolis, MN, USA, Apr. 1992.

[35] R. A. Harshman, "PARAFAC2: Mathematical and technical notes," *UCLA Work. Papers Phonetics*, vol. 22, pp. 30–44, 1972.

[36] H. A. L. Kiers, J. M. F. Ten Berge, and R. Bro, "PARAFAC2 -Part I. A direct fitting algorithm for the PARAFAC2 model," *J. Chemometrics*, vol. 13, no. 3-4, pp. 275–294, May 1999.

[37] P. H. Schoenemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.

[38] K. Naskovska, Y. Cheng, A. L. F. de Almeida, and M. Haardt, "Efficient computation of the PARAFAC2 decomposition via generalized tensor contractions," in *Proc. 2018 52nd Asilomar Conf. Signals, Syst., Comput.*, 2018, pp. 323–327.

[39] G. Favier and A. de Almeida, "Overview of constrained PARAFAC models," *EURASIP J. Adv. Signal Process.*, vol. 2014, pp. 1–25, Sep. 2014.

[40] F. Roemer and M. Haardt, "Tensor-based channel estimation and iterative refinements for two-way relaying with multiple antennas and spatial reuse," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5720–5735, Nov. 2010.

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

[42] B. Song, F. Roemer, and M. Haardt, "Flexible coordinated beamforming (FlexCoBF) algorithm for the downlink of multi-user MIMO systems," in *Proc. Int. ITG Workshop Smart Antennas*, 2010, pp. 414–420.

[43] B. Song, F. Roemer, and M. Haardt, "Flexible coordinated beamforming (FlexCoBF) for the downlink of multi-user MIMO systems in single and clustered multiple cells," *Elsevier Signal Process.*, vol. 93, no. 9, pp. 2462–2473, Sep. 2013.

Liana Khamidullina (Student Member, IEEE) received the bachelor's degree in telecommunications from Kazan National Research Technical University, Kazan, Russia, in 2016, and the M.Sc. degree in communications and signal processing from the German-Russian Institute of Advanced Technologies, Kazan, Russia, organized between Kazan National Research Technical University, and the Ilmenau University of Technology, Ilmenau, Germany, as a double degree program, in 2018. She is currently working toward the Ph.D. degree in communications and signal processing with the Ilmenau University of Technology.

Since 2019, she has been a Research Assistant with Communications Research Laboratory, Ilmenau University of Technology. Her research interests include tensor-based signal processing, multiuser MIMO precoding, radar, direction of arrival estimation, and biomedical signal processing.

André L.F. de Almeida (Senior Member, IEEE) received the double Ph.D. degree in sciences and teleinformatics engineering from the University of Nice, Sophia Antipolis, Nice, France, and the Federal University of Ceara, Fortaleza, Brazil, in 2007. He is currently a Professor with the Department of Teleinformatics Engineering, Federal University of Ceara. His research interests include channel estimation and equalization, multi antenna systems, blind and semi-blind signal processing, multilinear algebra, and tensor decompositions with applications to communications and signal processing. He was awarded multiple times Visiting Professor positions with the University of Nice Sophia-Antipolis (2012, 2013, 2015, 2018, and 2019).

He was an Associate Editor for several journals, such as the IEEE TRANSACTIONS ON SIGNAL PROCESSING (during 2012–2014 and 2014–2016), and the IEEE SIGNAL PROCESSING LETTERS (during 2016–2018 and 2018–2020). He is currently the Senior Area Editor of the IEEE SIGNAL PROCESSING LETTERS, and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. Prof. Almeida was an Elected Member of the IEEE SPS Sensor Array and Multichannel Technical Committee (during 2015–2018 and 2018–2021). He is an Elected Member of the IEEE Signal Processing Society Signal Processing Theory and Methods Technical Committee (during 2022–2024), and an Elected Member of the *EURASIP Signal Processing* for MultiSensor Systems Technical Area Committee (SPMuS TAC) (during 2016–2018 and 2019–2022). In 2021, he was elected as the Vice-Chair of the SPMuS TAC (during 2022–2023). He was also involved with the organization and chairing of several SPS conferences. In particular, he was a General Co-Chair of the 2017 IEEE International Workshop on Computational Advances in MultiSensor Adaptive Processing (CAMSAP'2017), Technical Co-Chair of IEEE GlobalSIP'2018 and IEEE GlobalSIP'2019 Symposia, Technical Co-Chair of the 11th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM'2020), and is the General Co-Chair of the IEEE CAMSAP'2023 workshop. He is the IEEE SPS Regional Director-at-Large for Regions 7 & 9 (during 2022–2023).

Martin Haardt (Fellow, IEEE) received the Diplom-Ingenieur (M.S.) degree from Ruhr-University Bochum, Bochum, Germany, in 1991, and the Doktor-Ingenieur (Ph.D.) degree from the Munich University of Technology, Munich, Germany, in 1996. Since 2001, he has been a Full Professor with the Department of Electrical Engineering and Information technology and the Head of the Communications Research Laboratory with the Ilmenau University of Technology, Ilmenau, Germany.

He studied electrical engineering with the Ruhr-University Bochum, Germany, and Purdue University, West Lafayette, IN, USA.

In 1997, he joint Siemens Mobile Networks in Munich, Germany, where he was responsible for strategic research for third generation mobile radio systems. From 1998 to 2001, he was the Director of the International Projects and University Cooperations in the mobile infrastructure business of Siemens in Munich, where his work focused on mobile communications beyond the third generation. His research interests include wireless communications, array signal processing, high-resolution parameter estimation, and tensor-based signal processing.

Prof. Haardt was the recipient of the 2009 Best Paper Award from the IEEE Signal Processing Society, Vodafone Innovations Award for outstanding research in mobile communications, ITG Best Paper Award from the Association of Electrical Engineering, Electronics, and Information Technology, and Rohde & Schwarz Outstanding Dissertation Award.

He was a Senior Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (since 2019), an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (during 2002–2006 and 2011–2015), IEEE SIGNAL PROCESSING LETTERS (during 2006–2010), the *Research Letters in Signal Processing* (during 2007–2009), the *Hindawi Journal of Electrical and Computer Engineering* (since 2009), the *EURASIP Signal Processing Journal* (during 2011–2014), and the Guest Editor of the *EURASIP Journal on Wireless Communications and Networking*.

From 2011 to 2019, he was an Elected Member of the Sensor Array and Multichannel technical committee of the IEEE Signal Processing Society, where he was the Vice Chair (during 2015–2016), Chair (2017–2018), and Past Chair (2019). Since 2020, he has been an Elected Member of the Signal Processing Theory and Methods technical committee of the IEEE Signal Processing Society.

Moreover, he was the Technical Co-Chair of the PIMRC 2005 in Berlin, Germany, ISWCS 2010 in York, U.K., the European Wireless 2014 in Barcelona, Spain, the Asilomar Conference on Signals, Systems, and Computers 2018, USA, and as the General Co-Chair of the WSA 2013 in Stuttgart, Germany, ISWCS 2013 in Ilmenau, Germany, CAMSAP 2013 in Saint Martin, French Antilles, WSA 2015 in Ilmenau, SAM 2016 in Rio de Janeiro, Brazil, CAMSAP 2017 in Curacao, Dutch Antilles, SAM 2020 in Hangzhou, China, the Asilomar Conference on Signals, Systems, and Computers 2021, USA.